Construyendo productos con FreeBSD

Resumen

El proyecto FreeBSD es un proyecto mundial, basado en el voluntariado y la colaboración, que desarrolla un sistema operativo portable y de alta calidad. El proyecto FreeBSD distribuye el código fuente de su producto bajo una licencia permisiva, con la intención de fomentar el uso de su código. Colaborar con el proyecto FreeBSD puede ayudar a las organizaciones a reducir su tiempo de comercialización, reducir los costes de ingeniería y mejorar la calidad de sus productos.

Este artículo examina los problemas al usar el código de FreeBSD en dispositivos y productos de software. Se destacan las características que convierten a FreeBSD en una excelente base para el desarrollo de productos. El artículo concluye sugiriendo algunas "mejores prácticas" para las organizaciones que colaboran con el proyecto FreeBSD.

Tabla de contenidos

1. Introducción 1
2. FreeBSD como un conjunto de bloques de construcción 3
3. Colaborando con FreeBSD 6
4. Conclusión 10
Bibliografía 10

1. Introducción

FreeBSD hoy es conocido como un sistema operativo de servidor de alto rendimiento. Se implementa en millones de servidores web y hosts conectados a Internet en todo el mundo. El código FreeBSD también forma parte integral de muchos productos, que van desde dispositivos como enrutadores de red, cortafuegos y dispositivos de almacenamiento, hasta computadoras personales. También se han utilizado partes de FreeBSD en software comercial empaquetado (ver FreeBSD como un conjunto de bloques de construcción).

En este artículo miramos al proyecto FreeBSD como un recurso de ingeniería de software - como una colección de bloques de construcción y procesos que puedes usar para construir tu producto.

Mientras que el código fuente de FreeBSD se distribuye libremente al público, para disfrutar plenamente de los beneficios del trabajo del proyecto, las organizaciones necesitan *colaborar* con el proyecto. En las secciones siguientes de este artículo tratamos de los medios eficaces para colaborar con el proyecto y los escollos que deben evitarse al hacerlo.

Aviso al Lector. El autor cree que las características del Proyecto FreeBSD enumeradas en este

artículo eran sustancialmente verdaderas en el momento en que se concibió y escribió el artículo (2005). Sin embargo, el lector debe tener en cuenta que las prácticas y los procesos utilizados por las comunidades de código abierto pueden cambiar con el tiempo y que, por lo tanto, la información de este artículo debe tomarse como indicativa en lugar de normativa.

1.1. Público objetivo

Este documento sería de interés para los siguientes grupos amplios de personas:

- Los responsables de la toma de decisiones en las empresas de productos que buscan formas de mejorar la calidad de sus productos, reducir su tiempo de comercialización y reducir los costes de ingeniería a largo plazo.
- Consultores de tecnología que buscan las mejores prácticas sacando ventaja del "open-source".
- Observadores de la industria interesados en comprender la dinámica de los proyectos de código abierto.
- Los desarrolladores de software que buscan usar FreeBSD y buscan formas de contribuir.

1.2. Objetivos del artículo

Después de leer este artículo, deberías tener:

- Comprensión de los objetivos del proyecto FreeBSD y su estructura organizativa.
- Comprensión de su modelo de desarrollo y procesos de ingeniería de versiones.
- Una comprensión de cómo los procesos convencionales de desarrollo de software corporativo difieren de los utilizados en el proyecto FreeBSD.
- Conocimiento de los canales de comunicación utilizados por el proyecto y del nivel de transparencia que puede esperar.
- Conocimiento de las formas óptimas de trabajar con el proyecto y de la mejor manera de reducir los costes de ingeniería, mejorar el tiempo de comercialización, gestionar las vulnerabilidades de seguridad y preservar la compatibilidad futura con su producto a medida que evolucione el proyecto FreeBSD.

1.3. Estructura del artículo

El resto del artículo está estructurado de la siguiente manera:

- FreeBSD como un conjunto de bloques de construcción hace una introducción del proyecto FreeBSD, explora su estructura organizativa, tecnologías clave y proceso de ingeniería de versiones.
- Colaborando con FreeBSD describe formas de colaborar con el proyecto FreeBSD. Examina fallos comunes encontrados por empleados de empresas trabajando con proyectos basados en voluntariado como FreeBSD.
- Conclusión conclusiones.

2. FreeBSD como un conjunto de bloques de construcción

FreeBSD constituye una excelente base sobre la que construir productos:

- El código fuente de FreeBSD se distribuye bajo una licencia BSD liberal que facilita su adopción en productos comerciales [Mon2005] con las mínimas molestias.
- El proyecto FreeBSD tiene excelentes prácticas de ingeniería que pueden ser aprovechadas.
- El proyecto ofrece una transparencia excepcional en su funcionamiento, permitiendo a las organizaciones que utilizan su código planificar eficazmente para el futuro.
- La cultura del proyecto FreeBSD, heredada del Grupo de Investigación en Ciencias de la Computación de la Universidad de California, Berkeley [McKu1999-1], fomenta el trabajo de alta calidad. Algunas características de FreeBSD definen la vanguardia de la técnica.

[GoldGab2005] examina con mayor detalle las razones comerciales para utilizar el código abierto. Para las organizaciones, los beneficios de usar componentes FreeBSD en sus productos incluyen un menor tiempo de lanzamiento al mercado, menores costes de desarrollo y menores riesgos de desarrollo.

2.1. Construyendo productos con FreeBSD

Estas son algunas de las formas en que las organizaciones han usado FreeBSD:

- Como una fuente de código probado para librerías y empresas de servicios públicos.

  Siendo un "derivado" del proyecto, las organizaciones aprovechan las nuevas características, arreglos de errores y pruebas que recibe el código base.

- Como un sistema operativo integrado (por ejemplo, para un enrutador OEM y un dispositivo de firewall). En este modelo, las organizaciones utilizan un kernel FreeBSD personalizado y un programa de aplicación junto con una capa de gestión propietaria para su dispositivo. Los OEM se benefician del nuevo soporte de hardware que agrega el proyecto FreeBSD en sentido ascendente y de las pruebas que recibe el sistema base.

  FreeBSD viene con un entorno de desarrollo auto alojado que permite crear fácilmente estas configuraciones.

- Como entorno compatible con Unix para las funciones de gestión de dispositivos de red y almacenamiento de alta gama, que se ejecutan en un procesador independiente "blade".

  FreeBSD proporciona las herramientas para crear imágenes dedicadas de sistemas operativos y programas. Su implementación de una API unix BSD es madura y está probada. FreeBSD también puede proporcionar un entorno estable de desarrollo cruzado para los otros componentes del dispositivo de gama alta.

- Como un producto para obtener amplias pruebas y soporte de un equipo mundial de desarrolladores para "propiedad intelectual" no crítica.

En este modelo, las organizaciones aportan infraestructura útil al proyecto FreeBSD (por ejemplo, ver netgraph(3)). La amplia exposición que obtiene el código ayuda a identificar rápidamente problemas de rendimiento y fallos. La participación de desarrolladores de alto nivel también conlleva extensiones útiles de la infraestructura de las cuales también se beneficia la organización que la aportó.

- Como entorno de desarrollo que soporta desarrollo cruzado para sistemas operativos empotrados como RTEMS y eCOS.

  Hay muchos entornos de desarrollo completamente equipados en la poderosa colección de 36000 aplicaciones portadas y empaquetadas con FreeBSD.

- Como una forma de soportar una API similar a Unix en un sistema operativo que de otro modo sería propietario, aumentando su atractivo para los desarrolladores de aplicaciones.

  Aquí partes del núcleo de FreeBSD y programas de aplicaciones son "portados" para ejecutarse junto con otras tareas en el sistema operativo propietario. La disponibilidad de una implementación del API Unix™ bien testeada y estable puede reducir el esfuerzo necesario para portar aplicaciones populares al sistema operativo propietario. Puesto que FreeBSD se proporciona con documentación de alta calidad para sus detalles internos y tiene una gestión efectiva de vulnerabilidades y procesos de ingeniería de versiones, los costes de mantenerse actualizado se mantienen bajos.

2.2. Tecnologías

Hay un gran número de tecnologías soportadas por el proyecto FreeBSD. A continuación se enumera una selección de ellos:

- Un sistema completo que puede auto alojarse de forma cruzada en muchas arquitecturas:
- Un núcleo modular con capacidad de multiprocesamiento simétrico, con módulos integrables en el núcleo y un sistema de configuración flexible y fácil de usar.
- Soporte para la emulación de binarios de Linux™ y SVR4 a velocidades próximas a la de la máquina. Soporte para binarios de drivers de red (NDIS) de Winwos™.
- Librerías para muchas tareas de programación: archivadores, soporte FTP y HTTP, soporte de hilos, además de un entorno de programación completo tipo POSIX™.
- Características de seguridad: Control de Acceso Obligatorio (mac(9)), jails (jail(2)), ACLs, y soporte para unidades criptográficas en el núcleo.
- Funciones de red: cortafuegos, gestión de la calidad de servicio (QoS), conexiones de red TCP/IP de alto rendimiento con soporte para muchas extensiones.

  La infraestructura de Netgraph (netgraph(4)) en el núcleo de FreeBSD permite a los módulos de red del núcleo conectarse conjuntamente de formas flexibles.

- Soporte para tecnologías de almacenamiento: Fibre Channel, SCSI, RAID software y hardware, ATA y SATA.

  FreeBSD soporta varios sistemas de ficheros y su sistema de ficheros nativo UFS2 soporta

actualizaciones blandas, instantáneas y tamaños de sistema de ficheros muy grandes (16TB por sistema de ficheros) [McKu1999].

La infraestructura GEOM (geom(4)) en el núcleo de FreeBSD permite a los módulos de almacenamiento del kernel configurarse de formas flexibles.

- Más de 36000 aplicaciones portadas, tanto comerciales como open-source, gestionadas mediante la colección de ports de FreeBSD.

2.3. Estructura de la organización

La estructura organizativa de FreeBSD no es jerárquica.

Esencialmente hay dos tipos de contribuyentes en FreeBSD, los usuarios generales de FreeBSD y los desarrolladores con permiso de escritura sobre el código fuente (conocidos como *committers* en la jerga).

Hay muchos miles de contribuyentes en el primer grupo; la gran mayoría de las contribuciones a FreeBSD provienen de individuos de este grupo. Los derechos de escritura (acceso de escritura) al repositorio se conceden a las personas que contribuyen de forma consistente al proyecto (denominados committers). Los derechos de escritura conllevan responsabilidades adicionales, y se asignan mentores a los nuevos committers para ayudarles a aprender lo que hay que hacer.

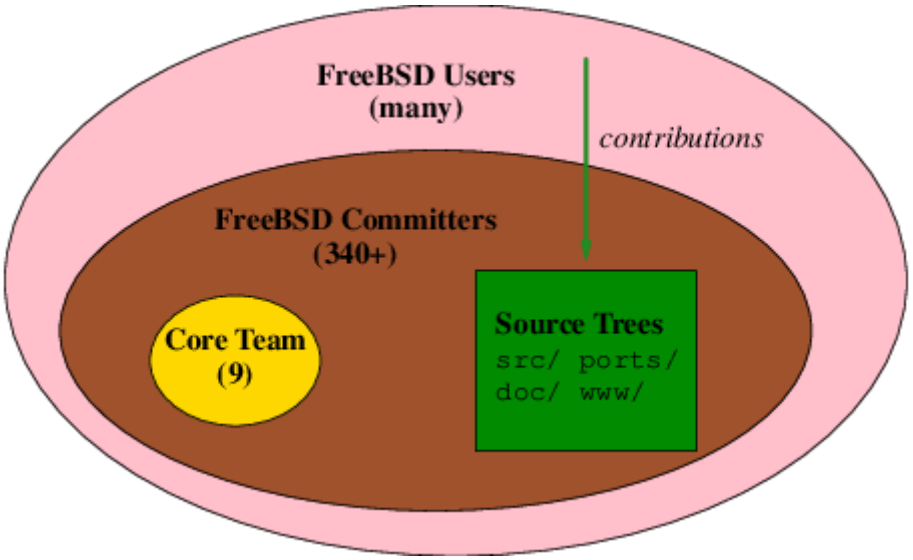

Figura 1. Organización de FreeBSD

La resolución del conflicto es llevada a cabo por un "Core Team" de nueve miembros que es escogido entre el grupo de committers.

FreeBSD no tiene committers "corporativos". A los committers individuales se les requiere que asuman la responsabilidad de los cambios que introducen en el código. La FreeBSD Committer's guide [ComGuide] documenta las reglas y responsabilidades de los committers.

El modelo de proyecto de FreeBSD se examina con detalle en [Nik2005].

2.4. Procesos de ingeniería de lanzamiento de versiones de FreeBSD

El proceso de lanzamiento de versiones de FreeBSD juega un papel importante en asegurar que las versiones liberadas son de alta calidad. En un momento dado, los voluntarios de FreeBSD soportan múltiples ramas de código (FreeBSD Release Branches):

- Características nuevas y código disruptivo entra en la rama de desarrollo, también conocida como la rama *-CURRENT*.
- Las ramas *-STABLE* son ramas de código que se obtienen de HEAD a intervalos regulares. Sólo el código probado se introduce en una rama -STABLE. Las características nuevas se añaden una vez que han sido probadas y estabilizadas en la rama -CURRENT.
- Las ramas *-RELEASE* son mantenidas por el equipo de seguridad de FreeBSD. Sólo se permiten arreglos críticos en las ramas -RELEASE.

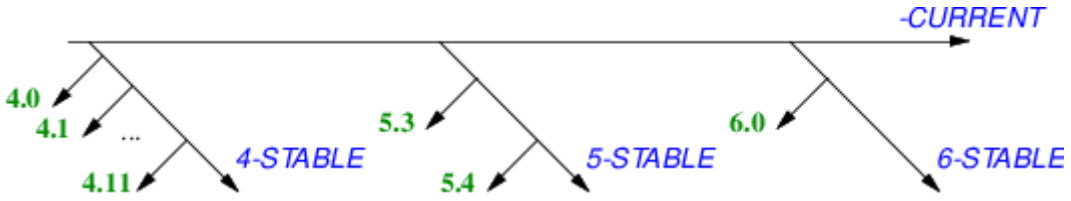

Figura 2. Ramas RELEASE de FreeBSD

Las líneas de código se mantienen vivas mientras haya interés en ellas por parte del usuario y del desarrollador.

Las arquitecturas se agrupan en "tiers"; Las arquitecturas *Tier 1* están soportadas por completo por los equipos de seguridad y de ingeniería de versiones del proyecto, las arquitecturas *Tier 2* están soportadas lo mejor posible y *Tier 3* está compuesto de arquitecturas experimentales. La lista de arquitecturas soportadas es parte de la colección de documentación de FreeBSD.

El equipo de ingeniería de versiones publica una hoja de ruta para futuras versiones de FreeBSD en el sitio web del proyecto. Las fechas que hay en la hoja de ruta no son fechas límite; FreeBSD se libera cuando el código y la documentación están listos.

Los procesos de ingeniería de versiones de FreeBSD se describen en [RelEngDoc].

3. Colaborando con FreeBSD

Los proyectos de código abierto como FreeBSD ofrecen código terminado de muy alta calidad.

Mientras que el acceso al código fuente de calidad puede reducir el coste del desarrollo inicial, a largo plazo los costes de gestión del cambio comienzan a dominar. A medida que los entornos informáticos cambian con el paso de los años y se descubren nuevas vulnerabilidades de seguridad, su producto también necesita cambiar y adaptarse. El uso de código abierto no se ve mejor como una actividad aislada, sino como un *proceso en curso*. Los mejores proyectos con los que se puede colaborar son los que son *vivos*; es decir, con una comunidad activa, objetivos claros y un estilo de trabajo transparente.

- FreeBSD tiene una comunidad de desarrolladores activa a su alrededor. En el momento de escribir este artículo hay muchos miles de contribuyentes de todos los continentes del mundo y más de 300 personas con acceso de escritura a los repositorios de fuentes del proyecto.
- Los objetivos del proyecto FreeBSD son [Hub1994]:
  - Desarrollar un sistema operativo de alta calidad para los equipos informáticos más utilizados, y,
  - Poner nuestro trabajo a disposición de todos bajo una licencia liberal.
- FreeBSD disfruta de una cultura de trabajo abierta y transparente. Prácticamente toda la discusión en el proyecto se produce por email, en listas públicas de correo que son archivadas para la posteridad. Las políticas del proyecto están documentadas y mantenidas bajo control de versiones. La participación en el proyecto está abierta a cualquiera.

3.1. Comprendiendo la cultura de FreeBSD

Para poder trabajar eficazmente con el proyecto FreeBSD necesita entender la cultura del proyecto.

Los proyectos impulsados por voluntarios operan bajo reglas diferentes a las de las corporaciones con fines de lucro. Un error común que cometen las empresas al aventurarse en el mundo del software libre es subestimar estas diferencias.

La mayoría de las contribuciones a FreeBSD se hacen voluntariamente sin que las recompensas económicas entren en escena. Los factores que motivan a los individuos son complejos, desde el altruismo hasta el interés por resolver los tipos de problemas que FreeBSD intenta resolver. En este entorno, la "elegancia nunca es opcional" [Nor1993].

La Visión a Largo Plazo. FreeBSD tiene sus raíces hace casi veinte años en el trabajo del Grupo de Investigación de Ciencias de la Computación de la Universidad de California en Berkeley.[1]. Cierto número de los desarrolladores originales del CSRG siguen asociados al proyecto.

Las perspectivas a largo plazo de los valores del proyecto [Nor2001]. Un acrónimo encontrado frecuentemente en el proyecto es DTRT, que significa "Do The Right Thing" ("haz lo correcto").

Procesos de Desarrollo. Los programas de ordenador son herramientas para la comunicación: a un nivel los programadores comunican sus intenciones utilizando una notación precisa a una herramienta (un compilador) que traduce sus instrucciones a código ejecutable. A otro nivel, la misma notación es utilizada para la comunicación de intenciones entre dos programadores.

Las especificaciones formales y los documentos de diseño son usados raramente en el proyecto. El código claro y bien escrito y los registros de cambio bien escritos ((Una entrada de log de cambio de ejemplo) se utilizan en su lugar. El desarrollo de FreeBSD sucede por "consenso aproximado y ejecutando código"[Carp1996].

```
r151864 | bde | 2005-10-29 09:34:50 -0700 (Sat, 29 Oct 2005) | 13 lines
Changed paths:
   M /head/lib/msun/src/e_rem_pio2f.c

Usa doble precisión para simplificar y optimizar la reducción de
```

```
argumentos también para argumentos de tamaño pequeño y medio:
en lugar de restar una aproximación float a pi/2 de 17+24, 17+17+24 o
17+17+17+24 bits, siempre resta un double de 33+53 bits. La versión
float es ahora más cercana a la versión double de lo que lo que era la
propia versión antigua -- utiliza la misma aproximación de 33+53 bits
como los casos más sencillos de la versión double, y cuando la versión
float tenía que cambiar al caso general lento |x| == 2^7*pi/2, ahora
cambia a |x| == 2^19*pi/2 al igual que la versión double.

Esto acelera la reducción de argumentos en un factor de 2 para |x| entre
3*pi/4 y 2^7*pi/4, y por un factor de 7 para |x| entre 2^7*pi/4 y 2^19*pi/4.
```

Ejemplo de una entrada de log de cambio

La comunicación entre los programadores es mejorada por el uso de un estándar común de codificación style(9).

Canales de Comunicación. Los colaboradores de FreeBSD están repartidos por todo el mundo. Email (y en menor medida, IRC) es el medio de comunicación preferido en el proyecto.

3.2. Buenas Prácticas para colaborar con el proyecto FreeBSD

A continuación presentamos algunas de las mejores prácticas para hacer el mejor uso de FreeBSD en el desarrollo de productos.

Planificar a largo plazo

Establecer procesos que ayuden a hacer el seguimiento del desarrollo de FreeBSD. Por ejemplo:

Seguimiento del código fuente de FreeBSD. El proyecto hace sencillo clonar su repositorio SVN utilizando svnsync. Tener el histórico completo de las fuentes es útil cuando se depuran problemas complejos y ofrece un conocimiento valioso de las intenciones de los desarrolladores originales. Utiliza un sistema de control de código capaz que te permita incorporar cambios entre el código base de FreeBSD y tu propio código.

Listado de código fuente anotado generado con `svn blame` muestra una porción de un listado anotado del fichero referenciado por el registro de cambio en <fig-change-log, Un ejemplo de entrada de registro de cambios>>. El antecesor de cada línea es claramente visible. Los listados anotados que muestran el histórico de cada fichero que forma parte de FreeBSD están disponibles en la web.

```
#REV         #WHO #DATE                                        #TEXT

176410        bde 2008-02-19 07:42:46 -0800 (Tue, 19 Feb 2008) #include
<sys/cdefs.h>
176410        bde 2008-02-19 07:42:46 -0800 (Tue, 19 Feb 2008)
__FBSDID("$FreeBSD$");
  2116        jkh 1994-08-19 02:40:01 -0700 (Fri, 19 Aug 1994)
  2116        jkh 1994-08-19 02:40:01 -0700 (Fri, 19 Aug 1994) /*
```

```
__ieee754_rem_pio2f(x,y)
  8870    rgrimes 1995-05-29 22:51:47 -0700 (Mon, 29 May 1995)  *
176552        bde 2008-02-25 05:33:20 -0800 (Mon, 25 Feb 2008)  * return the
remainder of x rem pi/2 in *y
176552        bde 2008-02-25 05:33:20 -0800 (Mon, 25 Feb 2008)  * use double
precision for everything except passing x
152535        bde 2005-11-16 18:20:04 -0800 (Wed, 16 Nov 2005)  * use
__kernel_rem_pio2() for large x
  2116        jkh 1994-08-19 02:40:01 -0700 (Fri, 19 Aug 1994)  */
  2116        jkh 1994-08-19 02:40:01 -0700 (Fri, 19 Aug 1994)
176465        bde 2008-02-22 07:55:14 -0800 (Fri, 22 Feb 2008) #include <float.h>
176465        bde 2008-02-22 07:55:14 -0800 (Fri, 22 Feb 2008)
  2116        jkh 1994-08-19 02:40:01 -0700 (Fri, 19 Aug 1994) #include "math.h"
```

Un listado de código anotado generado con `svn blame`

Usa un guardián. Designa un *guardián* para monitorizar el desarrollo de FreeBSD, para controlar los cambios que potencialmente podrían impactar en tus productos.

Reporta problemas. Si adviertes un fallo en el código de FreeBSD que estás utilizando, cumplimenta un reporte de error. Este paso asegura que no tengas que arreglar el fallo la próxima vez que actualices tu código con el original de FreeBSD.

Aprovecha los esfuerzos de ingeniería de versiones de FreeBSD

Utiliza código de una rama de desarrollo -STABLE de FreeBSD. Estas ramas de desarrollo están formalmente soportadas por los equipos de seguridad e ingeniería de lanzamiento de FreeBSD y se componen de código probado.

Donar código para reducir costes

Una parte importante de los costes asociados al desarrollo de productos es el mantenimiento. Al donar código no crítico al proyecto, te beneficias de que tu código tenga una exposición mucho más amplia de la que tendría de otro modo. Esto, a su vez, lleva a que se eliminen más errores y vulnerabilidades de seguridad y a que se identifiquen y solucionen las anomalías de rendimiento.

Obtén apoyo de manera efectiva

Para productos con fechas límite ajustadas, se recomienda que contrates o realices un acuerdo de consultoría con un desarrollador o empresa con experiencia en FreeBSD. Lista de correo sobre empleos relacionados con FreeBSD es un canal de comunicación útil para encontrar talento. El proyecto FreeBSD mantiene una galería de consultores y empresas de consultoría que realizan trabajo sobre FreeBSD. El BSD Certification Group ofrece certificaciones para todos los sistemas operativos importantes derivados de BSD.

Para necesidades menos críticas, puedes solicitar ayuda en las listas de correo del proyecto. Una guía útil para seguir cuando se solicita ayuda es [Ray2004].

Publicita tu participación

No es necesario que publicites tu uso de FreeBSD, pero hacerlo ayuda tanto a tu esfuerzo como al del proyecto.

Hacer saber a la comunidad FreeBSD que tu empresa utiliza FreeBSD ayuda a mejorar tus posibilidades de atraer talento de alta calidad. Una amplia lista de soporte a FreeBSD también significa más intercambio de ideas para los desarrolladores. Esto a su vez produce una base más saludable para su futuro.

Apoya a los desarrolladores de FreeBSD

A veces el camino más directo para introducir una nueva característica en FreeBSD es apoyar al desarrollador que ya está trabajando en un problema relacionado. La ayuda puede variar desde donaciones de hardware a ayuda financiera directa. En algunos países, las donaciones al proyecto FreeBSD se benefician de exenciones fiscales. El proyecto tiene un enlace de donaciones dedicado a ayudar a los donantes. El proyecto también mantiene una página web donde los desarrolladores indican sus necesidades.

Como política, el proyecto FreeBSD reconoce en su sitio web todas las contribuciones recibidas.

4. Conclusión

Los objetivos del proyecto FreeBSD son crear y obsequiar el código fuente de un sistema operativo de alta calidad. Trabajando con el proyecto FreeBSD puede reducir los costes de desarrollo y mejorar su tiempo de lanzamiento al mercado en una variedad de escenarios de desarrollo del producto.

Hemos examinado las características del proyecto FreeBSD que lo convierten en una excelente elección para formar parte de la estrategia de un producto de una organización. A continuación, examinamos la cultura predominante en el proyecto y las formas efectivas de interactuar con sus desarrolladores. El artículo ha concluido con una lista de buenas prácticas que podrían ayudar a las organizaciones que colaboran con el proyecto.

Bibliografía

[Carp1996] The Architectural Principles of the Internet B. Carpenter. The Internet Architecture Board.The Internet Architecture Board. Copyright® 1996.

[ComGuide] Committer's Guide The FreeBSD Project. Copyright® 2005.

[GoldGab2005] Innovation Happens Elsewhere: Open Source as Business Strategy Ron Goldman. Richard Gabriel. Copyright® 2005. Morgan-Kaufmann.

[Hub1994] Contributing to the FreeBSD Project Jordan Hubbard. Copyright® 1994-2005. The FreeBSD Project.

[McKu1999] Soft Updates: A Technique for Eliminating Most Synchronous Writes in the Fast Filesystem Kirk McKusick. Gregory Ganger. Copyright® 1999.

[McKu1999-1] Twenty Years of Berkeley Unix: From AT&T-Owned to Freely Redistributable Marshall Kirk McKusick. Open Sources: Voices from the Open Source Revolution O'Reilly Inc.. Copyright® 1993.

[Mon2005] Why you should use a BSD style license for your Open Source Project Bruce Montague. The FreeBSD Project. Copyright® 2005.

[Nik2005] A project model for the FreeBSD Project Niklas Saers. Copyright® 2005. The FreeBSD Project.

[Nor1993] Tutorial on Good Lisp Programming Style Peter Norvig. Kent Pitman. Copyright® 1993.

[Nor2001] Teach Yourself Programming in Ten Years Peter Norvig. Copyright® 2001.

[Ray2004] How to ask questions the smart way Eric Steven Raymond. Copyright® 2004.

[RelEngDoc] FreeBSD Release Engineering Murray Stokely. Copyright® 2001. The FreeBSD Project.

[1] El repositorio de código fuente de FreeBSD contiene la historia del proyecto desde sus comienzos y hay CDROMs disponibles que contienen código anterior del CSRG