

El Proyecto FreeBSD

Jesús Rodríguez

jesusr@freebsd.org

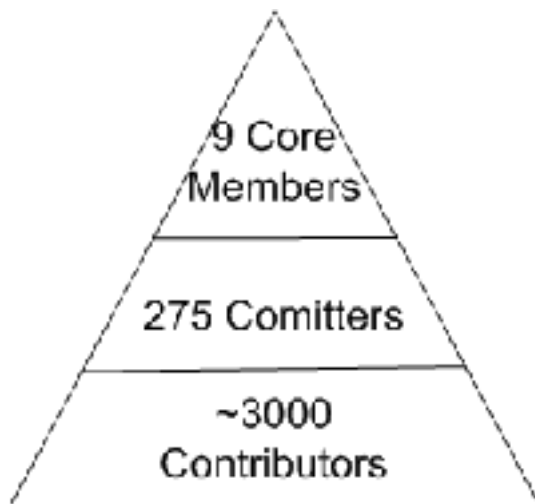
Copyright © 2003 por Jesús Rodríguez

\$Id: article.sgml,v 1.1.1.1 2003/05/22 22:45:29 jesusr Exp \$

Hasta ahora, el Proyecto FreeBSD ha publicado toda una serie de documentación descriptiva de cómo hacer diferentes trabajos y tareas pero no existí, y se ha hecho necesario, un documento que sumalice cómo se ha estructurado el proyecto debido al incremento en la incorporación de nuevos miembros ¹.

1. Estructura de organización

La estructura de organización de FreeBSD se ha dividido en los siguientes grupos para poder gestionar y coordinar de la mejor manera y más efectiva posible todos los recursos del proyecto: *core team*, *committers* y *contributors*:



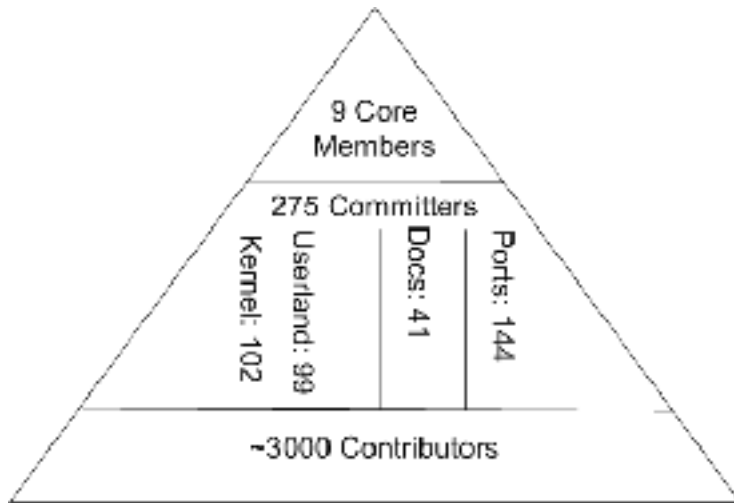
El recurso principal en la comunidad FreeBSD son sus desarrolladores: los *committers* y *contributors*. Es gracias a las contribuciones realizadas por estos dos grupos que el proyecto FreeBSD puede seguir avanzando. Con fecha del mes de mayo del 2003 se estima la participación de unos 3200 *contributors* en el proyecto.

Los *committers* son desarrolladores con acceso de escritura al código fuente original de FreeBSD. Normalmente, estos son los desarrolladores más activos e invierten su tiempo no sólo en trabajar en sus propias aportaciones de código si no en integrar el código enviado por otros desarrolladores que no tienen ese acceso de escritura (*contributors* o desarrolladores ocasionales). El grupo de *committers* también es el encargado de elegir mediante votación al *Core*

Team. El *Core Team* es un grupo de 9 *committers* encargado de dirigir el proyecto y son la máxima “autoridad” dentro del proyecto.

El proyecto puede agruparse en cuatro partes diferenciadas o subproyectos. Estos subproyectos son desarrollo del *kernel*, desarrollo de *user-land*, ports y documentación.

La segmentación comentada anteriormente modifica el triángulo de la siguiente manera:



El *kernel* es elemento principal sobre el que se basa FreeBSD. Mientras que las aplicaciones de *user-land* están protegidas contra fallos por otras aplicaciones, el sistema al completo es vulnerable a errores del *kernel*. Esto, combinado con la enorme cantidad de dependencias en el *kernel* y que no es fácil ver todas las consecuencias de un cambio en el *kernel*, hace que los desarrolladores que trabajan en esta área deban tener un relativo conocimiento del *kernel* en su totalidad.

Las utilidades principales, conocidas como *user-land* proveen el interfaz que identifica al interfaz de usuario, librerías compartidas e interfaces externos, además de la interacción con el *kernel*.

La documentación está gestionada por el *Proyecto de Documentación de FreeBSD* siendo responsables de toda la documentación incluyendo toda la información contenida en el servidor web de FreeBSD.

Los *Ports* son la colección de meta-datos necesaria para compilar correctamente los paquetes de software en FreeBSD. Un port, por ejemplo, es el navegador web Mozilla. El port contiene la información de donde obtener el código fuente, que parches aplicar y de que manera, y como instalar el programa en el sistema una vez compilado. Esto permite poder obtener el código fuente, compilarlo e instalarlo de manera totalmente automática. En la actualidad existen más de 8000 ports diferentes.

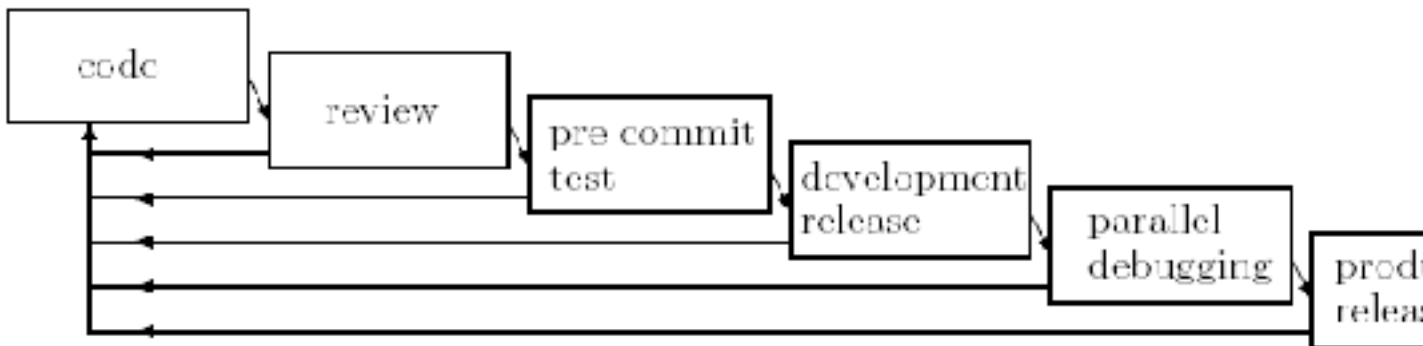
La mayoría de desarrolladores sólo estará integrado en una o como máximo dos de estas partes durante toda su participación y relación con el proyecto.

2. Modelo de metodología

2.1. Modelo de desarrollo

No existe un modelo definido de cómo los desarrolladores deben escribir código en FreeBSD pero sí se ha sugerido

un modelo de cómo el código es integrado en el proyecto:



La versión de desarrollo (development release) es la rama FreeBSD-CURRENT ("-CURRENT") y la versión de producción es la rama FreeBSD-STABLE ("-STABLE").

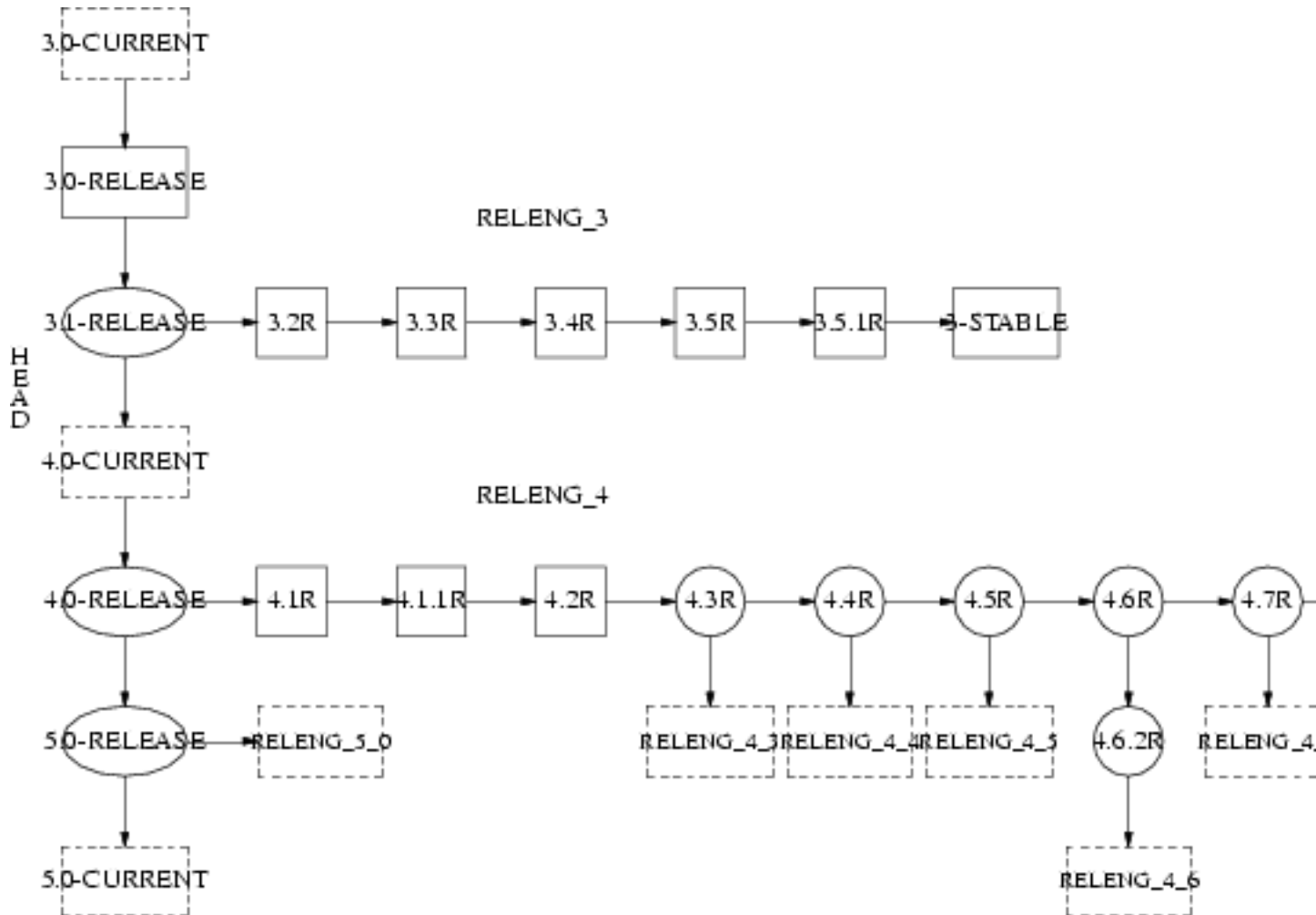
Este modelo de desarrollo es un modelo de cambio y muestra que después de escribir el código, los desarrolladores buscan y solicitan la revisión de ese código por parte de la comunidad. Una vez se integra el nuevo código en la versión de desarrollo de FreeBSD, los cambios son testeados por multitud de usuarios y desarrolladores. Después de verificar el correcto funcionamiento de los cambios, éstos son integrados en la versión de producción. Si cada una de estas etapas no finaliza de forma satisfactoria, el desarrollador debe volver atrás, realizar las modificaciones necesarias en el código y reiniciar el proceso.

La mayoría de desarrolladores de FreeBSD trabajan individualmente, lo que significa que el modelo descrito se usa de forma paralela por los diferentes desarrolladores trabajando en los diferentes desarrollos. Un desarrollador puede estar trabajando en múltiples cambios de forma que mientras espera a que cierto código sea revisado por los usuarios puede estar escribiendo o desarrollando otro código totalmente diferente. El hecho de integrar código en -CURRENT o -STABLE se llama "*hacer un commit*".

De diciembre del 2001 a diciembre del 2002 se realizaron un total de 132148 *commits* en FreeBSD. Esto se traduce en una media de 360 *commits* diarios, lo que da una idea de la gran actividad desarrollada en FreeBSD.

2.2. Ramas/Versiones

Para mejorar el entendimiento del modelo de versiones de FreeBSD, a continuación se muestra un esquema en forma de árbol en el que se describe el histórico de versiones. Las líneas que se siguen unas a otras en una dirección determinada representan una rama. Las cajas cuadradas y ovaladas con líneas continuas representan versiones oficiales. Las cajas con líneas punteadas representan las ramas de desarrollo en ese momento. Las ramas de seguridad se representan con cajas redondas. Los óvalos se diferencian de los cuadrados en que representan una separación (fork) en la que una rama se separa en dos ramas, siendo una de ellas una sub-rama. Por ejemplo, con la versión 4.0-RELEASE la rama 4.0-CURRENT se dividió en 4-STABLE y 5.0-CURRENT. Con la versión 4.5-RELEASE, esta rama se dividió en otra rama de seguridad llamada RELENG_4_5.



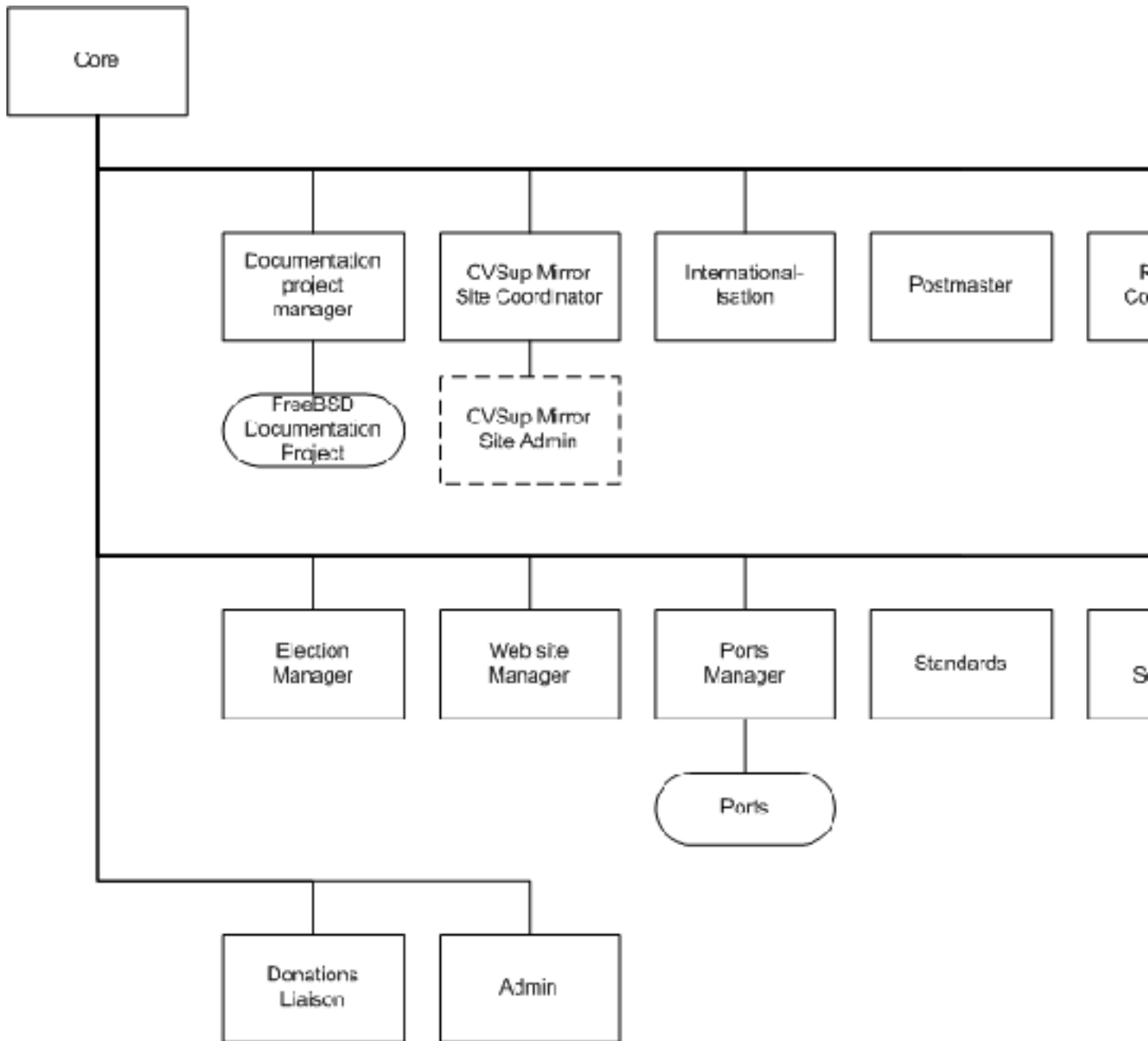
Una "major release" siempre se crea desde la rama -CURRENT. Aún así, la rama -CURRENT no tiene porque dividirse en ese punto en el tiempo y puede enfocarse en estabilizarse antes de llegar a la división. Un ejemplo de este modelo es que después de 3.0-RELEASE, la versión 3.1-RELEASE también fue una continuación de la rama -CURRENT y -CURRENT no volvió a ser una verdadera rama de desarrollo hasta se creó la rama 3-STABLE. Una vez -CURRENT vuelve a ser una rama de desarrollo sólo puede ser seguida de una "major release". La versión 5-STABLE se prevee que sea creada a partir de 5.0-CURRENT en la versión 5.1-RELEASE o 5.2-RELEASE. Hasta que no se cree la rama 5-STABLE, la rama de desarrollo no será marcada como 6.0-CURRENT.

2.3. Responsabilidades

Algunos *committers* tienen responsabilidades sobre algunas áreas en especial. Estas responsabilidades o roles se conocen como "hats" (sombreros). Estas responsabilidades pueden ser sobre funciones en el proyecto como relaciones públicas o responsable de una parte determinada y definida del código. Como este es un proyecto en el que los participantes aportan voluntariamente su tiempo libre, las personas con responsabilidades especiales asignadas no siempre están disponibles. Esto implica que los responsables deben nombrar a otras personas que puedan sustituirles en su ausencia o establecer que la responsabilidad es compartida por un grupo y no es exclusiva de una sólo persona.

Las responsabilidades *oficiales* del proyecto están más o menos formalizadas y corresponden prácticamente todas a

roles o posiciones administrativas. Tiene autoridad y responsabilidad absoluta sobre sus áreas. La siguiente ilustración muestra las líneas de responsabilidad:



2.4. Final

Esta ha sido una mínima introducción al modelo de desarrollo y estructura interna del Proyecto FreeBSD. Desde los inicios del proyecto hasta la actualidad se ha modificado enormemente el model de desarrollo debido, especialmente, al aumento de participantes y personas implicadas en el proyecto, debiendo establecer normas y reglas para intentar

asegurar una convivencia *pacífica* y fructífera para todos los participantes y seguidores del Proyecto FreeBSD.

Notas

1. A tener en cuenta la ley de Brook que indica que "añadir otra persona a un proyecto retardado hará que se retrase más". Un modelo de proyecto es una herramienta para reducir las necesidades de comunicación.