

Semana 2

Software Libre

Unidad 2

Ingenieria de software libre

Material compilado con fines académicos, se prohíbe su reproducción total o parcial sin la autorización de cada autor.

2. Ingeniería de software libre

2.1. La catedral y el bazar

Eric S. Raymond publicó en 1997 un artículo, el cual ha sido ampliamente difundido, “La catedral y el bazar”, donde se describe las características de los modelos de desarrollo de software libre. Este documento, hasta cierto punto, marcó el comienzo para el desarrollo de la ingeniería de software libre.

Raymond hace una analogía entre la manera de construir las catedrales medievales y la manera clásica de producir software. Indica que en ambos casos hay una evidente distribución de tareas, resaltando que hay un diseñador que está por encima de todo, el cual deberá controlar en todo momento el desarrollo de la actividad. También, se indica que la planificación está estrictamente controlada, dando lugar a procesos detallados en los que cada participante en la actividad tiene un rol muy delimitado.

A lo que Raymond indica como un modelo de creación de catedrales, además de tener cabida los procesos pesados, que es posible encontrar en la industria de software (el modelo clásico en cascada o las vertientes del RUP), también entran los proyectos de software libre, como es el caso de GNU y NetBSD. Para Raymond, estos proyectos se encuentran fuertemente centralizados, ya que pocas personas elaboran el diseño e implementación del software. Además, se indica que las entregas de estos programas están espaciados en el tiempo, siguiendo una planificación estricta. Por lo tanto, supone tener pocas entregas del software y ciclos largos entre las entregas y que constan de varias etapas.

Según Raymond, el modelo antagónico al de la catedral es el bazar. Algunos programas de software libre, especialmente el núcleo de Linux, han sido desarrollados siguiendo un esquema al de un bazar oriental.

- En un bazar no existe una máxima autoridad que controle los procesos que se están desarrollando, ni que planifique estrictamente lo que ha de ocurrir.
- Es muy posible que las actividades y roles de los participantes cambien de forma continua (los vendedores pueden pasar a ser clientes) y sin indicación externa.

Lo más novedoso de “La catedral y el bazar”, es la descripción de los procesos que ha hecho de Linux un éxito en el ámbito de software libre; teniendo una sucesión de buenas prácticas para aprovechar al máximo las posibilidades que permiten dis-

poner de código fuente e interactuar utilizando sistemas y herramientas telemáticas (González Barahona, 2010).

Raymond indica que en un proyecto de software libre se debe contar con una especie de líder, que generalmente coincide con el fundador del proyecto, el cual ha de guiar el proyecto y mantiene siempre la última palabra en la toma de decisiones. Las cualidades que debe tener este líder son principalmente:

- Saber motivar y coordinar un proyecto.
- Entender a los usuarios y codesarrolladores.
- Buscar acuerdos.
- Incluir a todo aquel que pueda aportar algo al proyecto.

¿Quieres saber más sobre Linus Torvalds y su trayectoria?



Visita la plataforma para ver el archivo relacionado

Si quieres saber más sobre “La catedral y el bazar”, pero desde la perspectiva de un desarrollador, distinto al autor del artículo.



Visita la plataforma para ver el archivo relacionado

A partir del crecimiento en tamaño y número de desarrolladores de algunos proyectos de software libre, han aparecido nuevas formas de administrar la toma de decisiones. En el caso de Linux, tiene una organización jerárquica basada en delegar responsabilidades por parte de Linus Torvalds, el líder del proyecto. En este caso, Linux cuenta con sus propios líderes, aunque éstos estén limitados por el hecho de que Linus Torvalds es el de la última palabra.

Este es un ejemplo claro de cómo la alta modularidad que existe en un proyecto de software libre ha derivado en una forma de organización y toma de decisiones específica.

2.2. Procesos en el software libre

Generalmente, el desarrollo de software libre implica un proceso menos formal que el llevado a cabo en el desarrollo del software propietario. La justificación radica en que, al ser un equipo de desarrollo trabajando en una modalidad libre, sus actividades son generalmente voluntarias o con recompensas económicas bajas, lo cual representa un compromiso poco estricto al trabajar con estos proyectos.

Enseguida se indican los procesos que se consideran más importantes en el desarrollo de software libre (EOi, 2012):

La forma de **capturar los requisitos** en el ámbito de software libre depende de la duración y del tamaño del proyecto. En las primeras etapas, el fundador y usuario del proyecto suelen ser una misma persona. Más adelante, y si el proyecto crece, el levantamiento de requisitos suele realizarse a través de las listas de correo electrónico, teniendo una clara distinción entre los desarrolladores y los usuarios. Para proyectos grandes, con muchos desarrolladores y usuarios, los requisitos son registrados usando la misma herramienta en la que se gestiona los errores del proyecto. En este caso no se habla de errores, sino de actividades, aunque el mecanismo es similar para ambos flujos.

Para ver el caso de HELLER, una herramienta libre para la especificación de Requisitos.



Visita la plataforma para ver el archivo relacionado

El **diseño global del sistema** sólo lo tienen los proyectos grandes, al realizar ampliamente la documentación del proyecto. En el resto de proyectos, es probable que quienes poseen este diseño sean los desarrolladores principales, en muchas ocasiones mentalmente o viéndose proyectado en versiones posteriores del software. La carencia de un diseño a detalle implica limitaciones en la reutilización de módulos desarrollados y además, los nuevos desarrolladores se enfrentan a un proceso lento de aprendizaje al informarse del desarrollo que se está efectuando.

La **implementación** es la etapa en la que se centra el mayor esfuerzo de los desarrolladores. Normalmente se sigue el paradigma de programación clásico de prueba y error, hasta que se consiguen los resultados deseados por el programador. También se usan las pruebas automáticas, donde existen máquinas dedicadas a descargar los repositorios y compilarlos en diferentes arquitecturas. Los errores detectados son notificados a los desarrolladores vía correo electrónico.

Si quieres conocer un caso de implementación de Software Libre.



Visita la plataforma para ver el archivo relacionado

Como se ha indicado previamente, la comunidad de software libre no posee una estandarización de tareas involucradas en el desarrollo. Sin embargo, los procesos indicados previamente dejan claro que sí existen ciertos procesos que los desarrolladores llevan a cabo para lograr cierta formalidad en sus desarrollos.

Hemos llegado al final del segundo bloque, en el cual, recapitulando, se ha dado a conocer la existencia del documento “La catedral y el bazar”, con el que se marca el comienzo para el desarrollo de la ingeniería de software libre. Recomendamos al alumno que para comprender ampliamente este tema, se dirijan al documento anexo en plataforma y hagan la lectura de la redacción elaborada por Eric S. Raymond. La segunda aportación del contenido está dada al exponer la diferencia entre el desarrollo de software libre y propietario. Esta información les ayudará en la solución de la actividad solicitada.