

El ecosistema del Software libre: Comunidades, Calidad, Coste.

Santiago Gala
the Apache Software Foundation
member
High Sierra Technology
Consultor



El Ponente

- Consultor en tecnologías de red, Internet, J2EE y Open Source desde 1996 (High Sierra Technology)
 - Committer en Jakarta desde 2000 (Apache Jetspeed)
- Miembro de la Apache Software Foundation desde 2003
 - Vice President, Apache Portals 2004-2007, PMC Jakarta, Labs
- Profesor en la Univ. Francisco de Vitoria
 - 2005-2007, enseñó IS/ADOO, Inteligencia Artificial, Servidores de Aplicaciones Web



Imágenes

- El dilema del Innovador
- ¿Es el Software Libre innovación disruptiva?
- Mitos sobre Software Libre
- Apache: 3 Recetas para comunidades
- Nuestro Reto

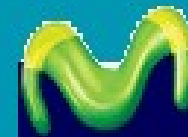


El dilema del Innovador



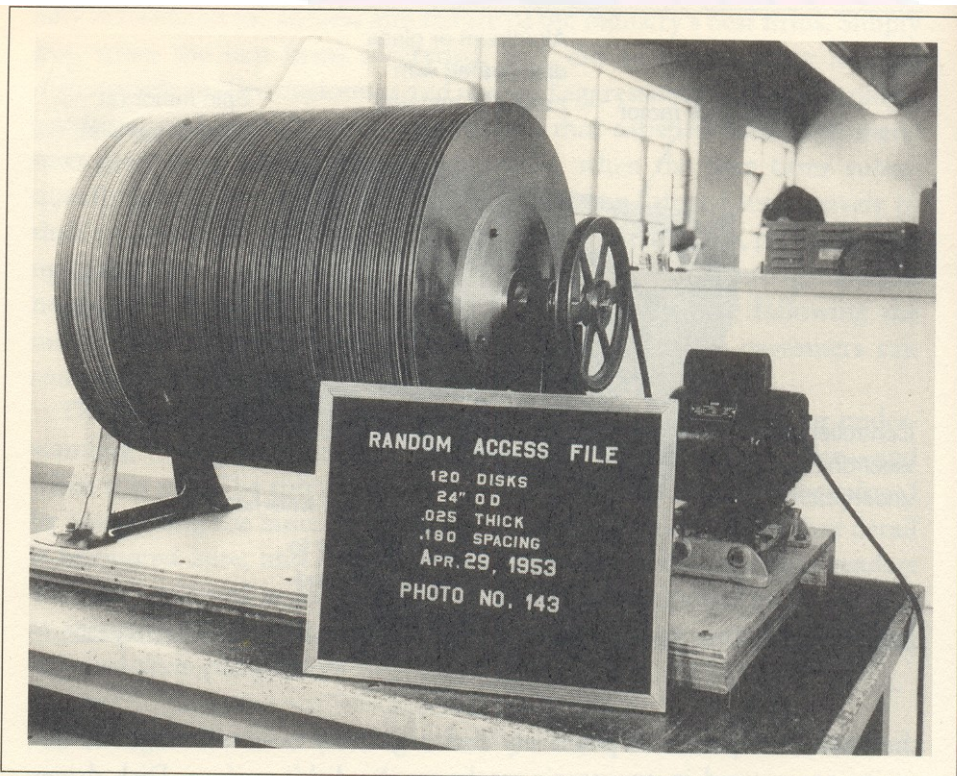
El dilema del innovador

- Clayton Christensen (1997)
 - Premio “Best Business Book” Financial Times 1997
- El dilema:
 - *Las decisiones de gestión, competentes y lógicas, que son esenciales para el éxito de una compañía son también las razones por las que ésta pierde su posición de liderazgo*
- ... en presencia de **tecnologías disruptivas**



Discos duros

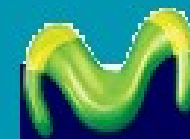
- Random Access Method for Accounting and Control, 5 MB en 50 discos de 24" (IBM San Jose, 1962)



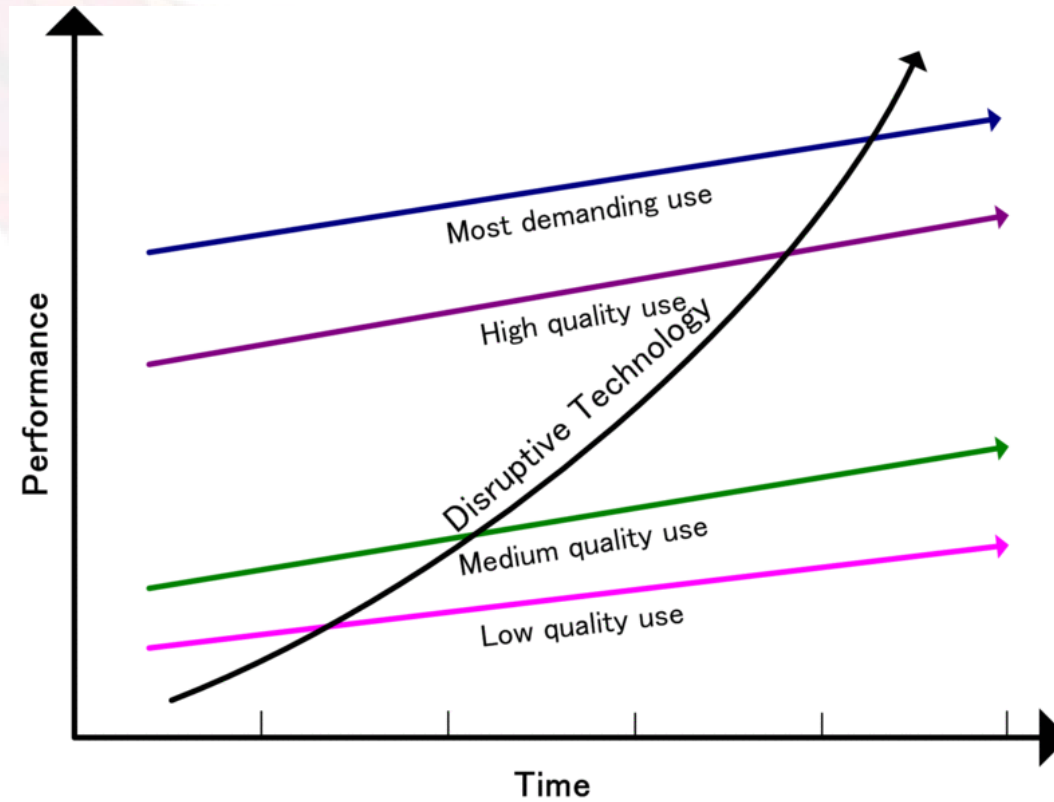
Source: Courtesy of International Business Machines Corporation.

¿Tecnologías disruptivas?

- Por sus bajas prestaciones y precio ó
- Por enfocarse a un nuevo tipo de clientes (o incluso a *non-consumers*)
 - wikipedia:
 - *Disruptive innovations can be broadly classified into lower-end and new-market disruptive innovations. A new-market disruptive innovation is often aimed at non-consumption, whereas a lower-end disruptive innovation is aimed at mainstream customers who were ignored by established companies.*



Comportamiento



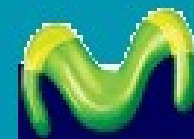
¿Es el software libre Tecnología disruptiva?



Los principios de la Innovación disruptiva:

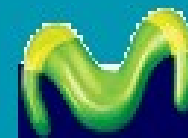
Aparece cuando/porque:

- Las Empresas dependen de sus clientes e inversores para conseguir recursos
 - negativa a ofrecer productos de bajo margen/prestaciones, y mejora *excesiva*
- Mercados Pequeños no justifican la necesidad de crecimiento de grandes empresas
 - Cuando el pequeño mercado tiene un crecimiento exponencial es ya tarde para entrar



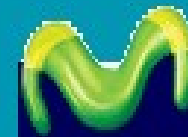
Los principios de la Innovación disruptiva (II):

- No se pueden analizar los mercados que no existen
 - Si el proceso de inversión exige cuantificar el mercado para entrar, se paralizan o cometen serios errores al enfrentarse a tecnologías disruptivas
- La oferta tecnológica sobrepasa la demanda de mercado
 - Prestaciones por encima de demanda del mercado: los clientes no basan su elección en las prestaciones. La elección evoluciona de funcionalidad a fiabilidad, a conveniencia y finalmente a precio.



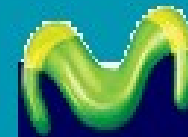
¿Tecnologías disruptivas en Software Libre?

- Inicialmente:
- Segmento especializado (new-market)
 - GNU toolchain, gcc, ..., mono
 - Linux kernel
 - Apache web server/LAMP/J2EE (***)
- Más cerca de Coste (low-end)
 - GIMP
 - OpenOffice.org, Linux Desktop
 - MySQL



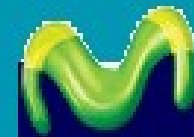
¿En qué es disruptivo el Software libre?

- El código fuente
 - permite la adaptación a (miles de) nichos
- El desarrollo abierto
 - reduce el coste exponencialmente
 - con retroalimentación de problemas resueltos, mejores diseños, bugs, fondo de inventario
 - a caballo de la Internet
- El ~~producto y sobre todo el~~ proceso de producción son innovación disruptiva



Pregunta para la reflexión

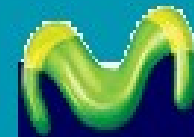
- ¿Es el software empresarial un producto en algún caso?
- Sólo el software “de consumo” o embebido se puede considerar producto
 - y sólo cuando no se busca su mejora continua
- Nuestro producto es el proceso de desarrollo: proyectos como focos de esfuerzo



¿Peores productos que con el viejo modelo?



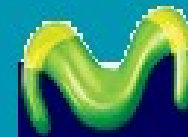
- Sí (inicialmente descoordinación)
 - pero se mejora exponencialmente
 - y se escala mucho más allá
- ¡Ojo!, nuestros productos son procesos
- Arthur C. Clarke:
 - *El ojo es una cámara fotográfica bastante mediocre... si no olvidamos que está hecha de gelatina.*



movistar

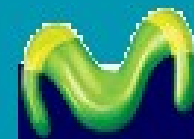
¿en qué es peor?

- ¿es eXtreme Programming disruptivo? ¿en qué es peor que el estilo convencional?
 - *XP es peor porque no permite la ilusión de control por parte de los gestores. La “pérdida de control” es uno de los costes de adoptar XP, y los principales clientes son los gestores. (c2.com)*
- **Tesis:** El proceso abierto hace al software libre peor y, por tanto, mejor.



¿peor?

- Un libro de bolsillo es peor que uno miniado en oro y copiado en un monasterio benedictino
- Un plato duralex es peor que una vajilla hecha a mano
- Un PC es peor que una Sun Enterprise



Mitos sobre Software Libre



Hechos / Mitos

- El software libre es profesional
 - La mayor parte de los desarrolladores tienen un curriculum abultado
 - ... y sus errores los puede ver cualquiera
- El software libre puede ser comercial
 - La mayor parte se desarrolla en compañías como IBM, Sun, Nokia, MySQL



Hechos / Mitos

- El software libre no es negocio
 - Esas compañías no son conocidas por tirar el dinero
- Sólo se hace en la Universidad y PyMEs
 - Hay pequeñas y grandes compañías, consultores y universidades
 - Un nuevo modelo de desarrollo
- El software libre es gratis
 - Se puede ganar dinero y cobrar por software libre



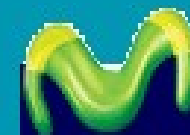
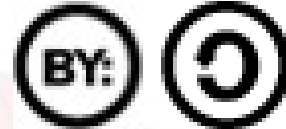
Mitos sobre licencias

- Todo software se recibe con licencias que limitan su uso, también el software libre
 - El software libre hace **explícitas** las limitaciones
- En casi todos los casos de uso, las licencias libres no limitan el negocio
- Lo que **no funciona** es vender copias sin valor añadido



Licencias

- Muchas Licencias:
 - Licencias BSD, MIT, MPL, Apache License
 - GPLv2... y ahora GPLv3
 - LGPL
- consulte con su abogado



Reglas sobre Licencias

- Enlazar con código GPL obliga a entregar los fuentes
- Se puede usar LGPL para evitar ese efecto mientras se mantiene abierto el código
- Las licencias BSD, MIT, MPL, Apache License permiten enlazar código propietario con ellas
- Se puede usar código GPL+propietario si se comunica mediante red, dbus, etc.



Licencias

- ¡Ojo con provisiones de Licencia de Patentes!
(AL2.0, GPLv3)
- Uso como servicios de red (GPLv3)
- <http://fsf.org>
- <http://opensource.org>
- <http://creativecommons.org>

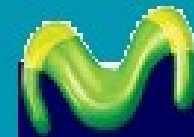


Apache: recetas para comunidades



Las jerarquías hacen agua

- No pueden gestionar
 - la evolución acelerada (seguridad, cambios)
 - a un coste razonable
- ¿por qué?
 - la coordinación es un cuello de botella
 - demasiada complejidad para “fabricar”
- Hay que abandonar la “fantasía de control”
 - dejar que el software **crezca**



La historia de Apache

- A partir de un código abandonado (NCSA web server)
- Parches en comunidad de usuarios
 - Brian Behlendorf organiza
 - Un grupo de una docena de webmasters
- El desarrollo sigue en la ASF
- 70% de cuota de mercado
 - Resistiendo dos ataques de Microsoft (2001-2002)
- Apache 2.0 es una gran base para desarrollos en red (LAMP)



¿Comunidad?

Good ideas
+ Bad code
= **Communities**

Stefano Mazzocchi (ASF Board)



¿Qué es un proyecto?

- Un proyecto necesita:
 - Una **Visión**
 - Satisfacer una **necesidad** (the *itch*)
 - **Usuarios** a los que les guste cómo funciona (sea por simplicidad de diseño, economía de recursos, estabilidad, costo, o incluso features ;-)
 - **Gestión** para mantener su evolución y evitar que se deshaga en trozos en el contraste entre las necesidades de los usuarios y las necesidades del diseño



... Y un poco de código (*)

- Una marca que haga fácil para la gente recordar que existe
- Documentación
- Un **equipo** de desarrolladores que depure, documente y reestructure el código continuamente
- ... y un poco de **código**.

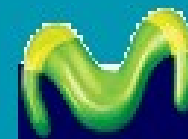
Santiago Gala (jakarta-general, 2003)



Discurso de nominación (Sam Ruby)



- Lo que más me gusta de la ASF es que no es una gran empresa. No es un sitio donde haga falta "proceso de firmas". Es un sitio donde la delegación y el "empowerment" no son puros eslóganes de marketing. **Las cosas se suelen equilibrar solas** en la ASF: las cosas que hay que hacer (como liberar versiones) no encuentran escasez de voluntarios. Las cosas que necesitan tiempo, como la incubación y la creación de PMC (Project Management Committees), se toman su tiempo.
- **Valoro la comunidad, el código y la arquitectura, en ese orden.** Me gusta trabajar con gente que valora las mismas cosas en orden diferente.



movistar

Nuestro Reto



Nuestro reto como consultores

- Llevar esos métodos a las empresas/organizaciones
- No intentar **fabricar árboles**: crecen solos
 - si el entorno es adecuado
- Abrir los procesos a la transparencia y la participación



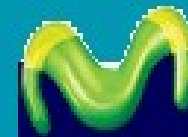
Nuestro reto (Movistar):

- Eliminar las barreras de entrada
 - autoprovisión, librerías, ejemplos
- Exponer los sistemas core
 - Igual que la web -> servicios web



Técnicas para empezar

- Construir usando módulos OS, y hacer comprender a los gestores que deben “gestionar las diferencias”
 - cultura del parche
 - gestión de versiones
 - integración continua



Resumen

- Si no se participa en la comunidad, es como golpear una pared
 - code rot
 - atropellados por versiones/características
- **Be community, my friend!**
- Participar no es (sólo) por ética, es necesidad de negocio

