# Open Source: Making A Business Case

## Deciding to join the gentle revolution

Phil Smith III
Levanta, Inc.

SHARE 103 / EITM
August 2004
Sessions 7019 / 9201

# Copyright Information

# Disclaimer

No animals were injured in creating this presentation.  Void where prohibited by law. All rights reserved. Your mileage may vary.  Available while quantities last. Used with permission.  Presentation was current at time of printing.  Abandon hope all ye who enter here.  For research purposes only.  Any resemblance to real persons, living or dead, is purely coincidental.  Presenter does not carry cash. Limitations on coverage and remedies apply.  Formatted to fit your screen. Please remain seated until the presenter has come to a complete stop.  All names listed are proprietary trademarks of their respective corporations.  Use only as directed.  No purchase necessary.  Must be over 18. Avoid contact with skin.  May be too intense for some viewers.  Some restrictions may apply. Not affiliated with the American Red Cross. Not responsible for direct, indirect, incidental, or consequential damages resulting from any defect, error, or failure to perform. This is not an offer to sell securities.  Views expressed may not be those of the sponsor. No other warranty expressed or implied.  Contains a substantial amount of non-tobacco ingredients. Inspired by a true story. Not responsible for typographical errors.  Specifications subject to change without notice. Prerecorded for this time zone. All models over 18 years of age.

# Agenda

- Why Open Source?
- What About Bill?
- Understanding the Issues
- Making the Decision

# Open Source Defined

## The Open Source definition – OpenSource.org

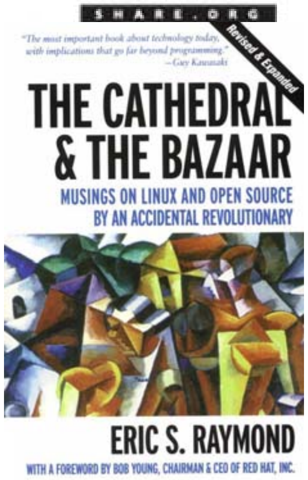
- Free redistribution required
- Source code provided
- Derived works must be allowed
- Integrity of the author’s source code may be controlled
- No discrimination against persons, groups, fields of endeavor
- Distribution of license must transfer with the code
- License must not be product-specific or restrict other software

## Open Source Software (OSS) Examples

- Linux
- Samba
- Apache
- Sendmail (some versions)

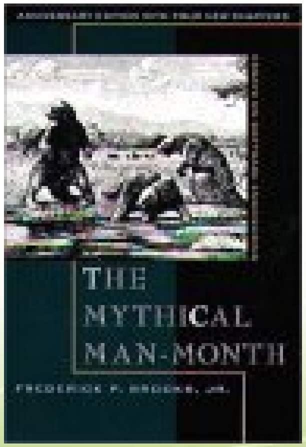
# Required Reading

- The Cathedral and the Bazaar
  - Eric S. Raymond ISBN 0-596-00108-8


"The most important book about technology today, with implications that go far beyond programming." —Guy Kawasaki
Revised & Expanded
THE CATHEDRAL & THE BAZAAR
MUSINGS ON LINUX AND OPEN SOURCE BY AN ACCIDENTAL REVOLUTIONARY
ERIC S. RAYMOND
WITH A FOREWORD BY BOB YOUNG, CHAIRMAN & CEO OF RED HAT, INC.

- "In The Beginning was the Command Line"
  - Neal Stephenson  
    www.cryptonomicon.com/beginning.html

- The Mythical Man-Month
  - Fred Brooks ISBN 0-201-83595-9



# Open Source Characteristics

- (Usually) platform independent
- UNIX application compatible
- Standards-based
  - Multi-vendor support
- Highly skilled, dedicated developers
- Source code provided
  - Open inspection of all functions possible
  - Extensible

# Why Open Source?

# The Short Answer

- Analgesia:
  - Management looking for ways to cut TCO
  - Staff tired of waiting on hold for support
  - Vendors cannot afford to build new applications from scratch
  - Developers hate reinventing the wheel

# The Shortest Answer

- Microsoft
- …or Sun microsystems
- …or ORACLE
- …or ca
- …or IBM
- …or whichever vendor you love to hate!

# The Longer Answer

- Multiple, often wildly divergent perspectives:

  1) Customer staff
  2) Customer management
  3) Vendor staff
  4) Vendor management

- These are ***not*** the same constituencies!
  - Understanding this is essential to understanding (and countering) arguments

# Customer Staff: Pro

- Source provided
  - Can understand and fix problems
- Standards-based
  - Proper behavior (at least somewhat) defined
- Publicly supported
  - Lots of others to give help with problems
  - “Community” development aspect is appealing
- Exploits high-end skills

# Customer Staff: Con

- Source provided
  - “I don’t want to fix someone else’s problems!”
- Standards-based
  - RFCs can be hard to understand
- Publicly supported
  - Documentation often poor or nonexistent
- May not have high-end skills to exploit
  - “I like my Microsoft GUI tools”

# Customer Management: Pro

- Low- or no-cost
  - Implies lower TCO
- Standards-based
  - Interoperability removes vendor lock-in
- Trendy—touted by trade rags and airline magazines
- Openness forces vendors toward interfaces etc. that customers actually want and use

# Customer Management: Con

- Source provided
  - Hackers can read the code, find vulnerabilities
  - “I don’t pay staff to fix others’ problems”
  - “My business can’t be dependent on local mods”
- Not formally supported
  - “I can’t bet my job on some kid in Finland”
- Trendy
  - “Management by magazine” isn’t management
- Open protocols aren’t necessarily better

# Vendor Staff: Pro

- Source provided
  - No dependency on someone else’s stuff
  - Can jump-start product development
- Standards-based
  - Appeals to anti-corporate sentiments
- Publicly supported
  - Even vendor engineers know they don’t know everything!

# Vendor Staff: Con

- Source provided
  - I’m not giving away my code!
- Standards-based
  - Long-haired geeks writing RFCs aren’t businesspeople, don’t know what’s truly needed
- Publicly supported
  - There’s no formal specification
  - What will I do when development forks?

# Vendor Management: Pro

- Customer demand
  - Customers are moving to Open Source
- Rapid Application Development (RAD)
  - Building on Open Source applications saves big development $$$ and time
- Openness promotes exploitation
  - Add-ons, ideas, enhancements (cf. id Software’s game Doom)
- Facilitates partnering with other vendors
  - Work from the same “playbook” (RFC)

# Vendor Management: Con

- We’re a business
  - “How can we charge thousands of dollars for something that we give away?”
- Intellectual property protection
  - “GPL means we have to give it away”
- “Proven” failure of model:
  - VA Linux, Netscape, et al.
  - “They’re all gone”

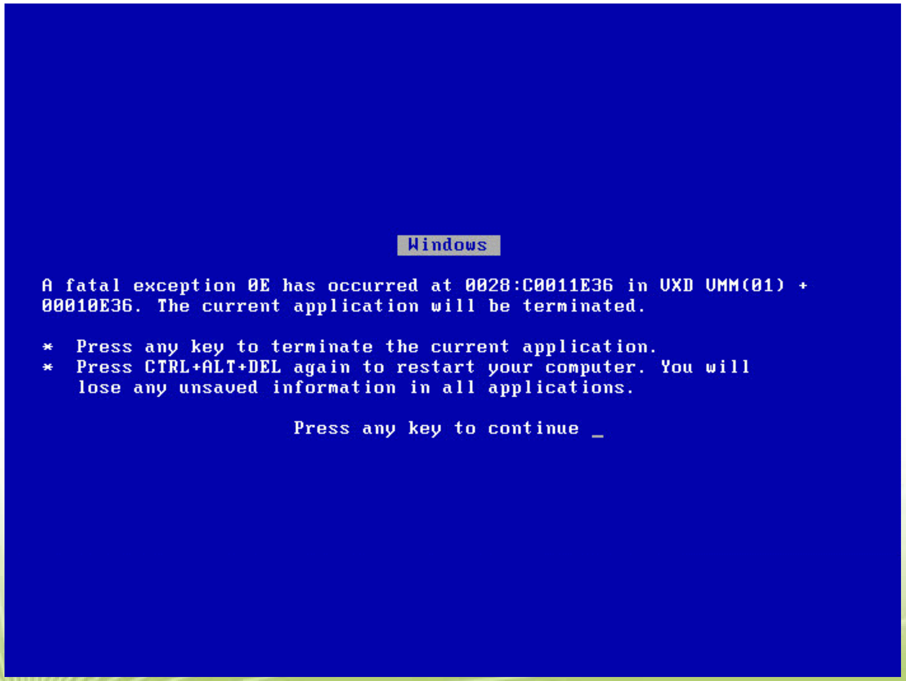
# What About Bill?

# Microsoft vs. Open Source

- Many Open Source advocates openly hate Microsoft
  - But emotional arguments are not business cases!
  - Many fail to recognize their emotional involvement
- Bad for Open Source and the community
  - Focus on real needs, avoid histrionics and invective
- Desktop and server considerations differ
  - Many Open Source apps exist for both
  - Windows is by far the leading desktop platform, users want it
  - Telling them “You’re wrong” isn’t productive

# Microsoft Is Not Evil

- Redmond is a business
  - Has provided excellent return to stockholders
- Products are carefully designed and developed
  - Tens of millions of users love them!
- Consumers’ willingness to put up with BSODs, etc. is not Bill’s fault
- Apparently the value of Windows is greater (for most) than the pain



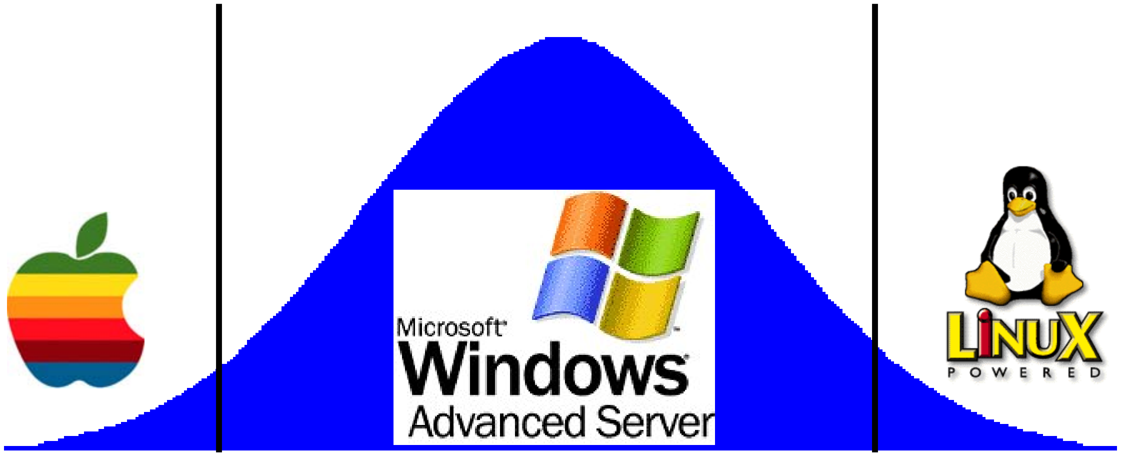
```
                              Windows

A fatal exception 0E has occurred at 0028:C0011E36 in VXD VMM(01) +
00010E36. The current application will be terminated.

*  Press any key to terminate the current application.
*  Press CTRL+ALT+DEL again to restart your computer. You will
   lose any unsaved information in all applications.

                       Press any key to continue _
```

# Microsoft Isn’t Perfect

- De facto monopoly position may reduce quality
  - Lack of competition means less corporate incentive toward truly excellent quality
  - This isn’t “evil”, it’s a business reality
- Internet service distribution (Windows Update) greatly reduces service costs
  - May further dampen initial product quality

# The Reality

- Most folks think Windows is “good enough” (server or desktop!)
- Classic bell curve distribution:



- We can argue about where the lines should be, but this essentially reflects the reality — ***today***

# Microsoft Has A Problem

- Microsoft's business model depends on customers upgrading to newer versions
  - Open Source applications threaten its ability to cram upgrades down consumers' throats
- PCs have passed the point where newest, fastest is necessary for reasonable use
  - Increasing consumer resistance to upgrades
- Anti-trust issues are a huge distraction

# Microsoft Is Not Stupid

- “Embrace and conquer” works
  - Just ask Novell, WordPerfect, Netscape…
- Integration is the key
  - Love ’em or hate ’em, Microsoft applications work together better than a mishmash
  - MMC “Snap-ins”, (moderately) consistent interfaces beat out command lines with most folks
- They are ***not*** ignoring Linux, Open Source!
  - www.opensource.org/halloween/
    (old but still interesting)
  - Microsoft attends LinuxWorld et al. nowadays

# Don’t Count Microsoft Out…

- They can react quickly—remember their 1995 turnaround on the Internet!
  - Consider their current “security focus” sparked by consumer confidence issues (and antitrust)
- Prediction: debugging tools on the horizon
  - First sign: Internet Explorer error reporting, which sends ABEND information to Microsoft
  - Now: Windows XP error reporting, extending to more applications

# Understanding the Issues

# Why Do We Have Computers?

- You don’t buy computers to run OSes
  - Applications provide value, ROI
- Operating Systems are a dead end without new applications
  - At some point you ***must*** upgrade
- But “it works well enough” is compelling!
  - DOS, Windows 3.1/95, old Macs in daily use

**Microsoft**

*has combined the strengths of its three most powerful operating systems to create its next generation operating system.*

Microsoft Windows CE

Microsoft Windows Me Millennium Edition

Windows NT Workstation

Microsoft Windows CEMeNT

*As hard as a rock and as dumb as a brick*

# Return With Us Now…

- …to those thrilling days of yesteryear:
- Most applications written in-house
- Staff retention recognized as important to preserve “institutional knowledge”
- Staff ability to react to problems critical to survival!
- Vendors were partners, not adversaries

# Why Was That Good?

- Detailed staff knowledge of internal applications was considered competitive advantage
- Small, controlled development followed Brooks’ Law (“Adding manpower to a late software project makes it later”, aka “Complexity and bugs rise with the square of the number of programmers involved”)
- Intangible but real: Staff “big picture” vision enabled avoiding some stupidity (cf. CRM disasters…)

# How Is Open Source Different?

- “Closed” theology appears to conflict with Open Source
  - But self-destiny huge advantage, then and now
- Brooks’ Law seems not to apply:
  - “Given enough eyeballs, all bugs are shallow”
- “Hacker culture” fundamentally different from traditional development culture
  - “Gift culture” makes knowledge-sharing valuable
  - Contrast with “proprietary advantage” theology
- As technology matures, advantage is how it gets applied, not what you have

# The Car–Linux Analogy

- Stephenson writes of:
  - Windows: Station wagons – ugly but popular
  - Apple: Euro-styled sedans – sexy but unpopular
  - Be: Batmobiles – very cool but hardly sellable
  - Linux: M1 Abrams tanks

- “I don’t know how to maintain a tank!”
  - “You don’t know how to maintain a station wagon, either!”

# The Car–Linux Analogy

- Applicable to any Open Source vs. vendor application
- Speaks to Windows' (apparent) simplicity and slickness vs. Linux's historical lack thereof
- (Though Linux isn't hard to get/install/use nowadays!)
- Vibrancy of Open Source community appeals to programmers and users
    - Many folks who will never use source appreciate when it's available
    - Provides emotional connection lacking in "slick" Windows and Windows products
    - Consider shareware's success and rabid fan base

# Everyone Looks Spiffy With A Tux

from `http://www.ubergeek.tv/switchlinux/`

# Making The Decision

Is Open Source right for my company?

# When Open Source?

- Open Source proven effective when use is planned, defined, understood
  - E.g., distributed/infrastructure servers
  - Linux, Samba, Apache, Sendmail, etc.
- Clearly evolving
  - More applications, less infrastructure than a year ago
  - WebSphere, DB2 Connect, WebLogic, et al.
  - Good sign for Open Source: shows maturation

# When Not Open Source?

- Integration and customization issues
  - Vendor apps typically better integrated (of course, “Integrated” may mean “We put it all on one CD”)
  - Some types of applications “always” require significant custom work (e.g., CRM systems)
  - OSS versions may require more local expertise
  - On the other hand, OSS means you can do the work yourself—avoid paying con$ulting fee$

# When Not Open Source?

- **Sunk costs:**
  Existing, paid for product licenses
  - Consider switching at next upgrade cycle

- **Training and conversion:**
  Costs can be surprisingly high
  - Some groups may rely on product features unknown to IT staff/upper management

# Choosing Open Source

- **Is the Open Source app good enough?**
  (Is it even close?)
  - Can you wait for it to get there?
  - Is commercial alternative good enough?
- **Is Open Source direction rational?**
  - Not just a reaction to dislike of a vendor
- **Is self-destiny benefit/avoidance of risk worth potential internal support cost?**

# Open Source vs. Vendor Apps

- Open Source typically more secure than vendor code
  - Reading source exposes weaknesses
  - Availability of fixes often measured in minutes
- Traditional vendors provide support “guarantees”
  - Can they live up to them?
  - If they don’t, what remedies do you have?
- Mission-critical systems/applications require serious support—no question
  - Serious support is available for OSS these days
  - And if that fails, you can still fix it yourself!

# Open Source vs. Vendor Apps

- Depending on platform, commercial product fixes may be essentially unavailable anyway
  - IBM (mostly) still gets it right
  - Have you ever gotten Microsoft to write a Windows patch for you?
- Commercial, closed applications are rarely more than 80% "done"
  - Insufficient ROI from further development

# Open Source Is Evolving

- Service and support available
  - RedHat, SuSE (United Linux), IBM, etc.

- Support, participation, and investment by major vendors
  - IBM, Sun, HP, Dell, etc.

- Open Source Development Lab (OSDL) projects targeting new, specific customer sets
  - Carrier Grade Linux, Data Center Linux

# The Real OSS Motivator(s)

- Saving money
- Saving time (which is really money)
- Saving staff (which is really money)
- Improving RAS (which saves money)
- Improving functionality (which saves money)

# How Do You Choose?

- Where are your real costs?
- Cost breakdown, biggest to smallest:
  - Labor: sysprogs, operators, et al.
  - Facilities
  - Hardware
  - Software (increasing mostly due to ISVs)
- How do you control TCO?

# Controlling TCO

- Obvious answer: stabilize/reduce spiraling costs
- Open Source can often help:
  - Labor: Many Open Source apps are very mature
    - As applications/systems increase, the same number of people can continue to support them
  - Facilities: Server consolidation can save big
    - More stuff on fewer boxes
  - Hardware: Server consolidation again
    - Stop wasting hardware for theoretical peak load!
  - Software: The most obvious opportunity
    - Things are tough all over – ISVs aren't cheap

# Summary

# Open Source Value

- Increased choice and power
  - True portability: hardware, operating system, apps Increased competition
  - Increased heterogeneity
  - Platform substitution and commoditization
- Available skills
  - Highly skilled developers
  - Large skill pool of employees
- Low- or no-cost for the function
- Trendy and hyped

# Summary

- Primary Open Source drivers should be financial
  - True cost/benefit of switching requires analysis
  - Emotional arguments need not apply
- But include intangibles
  - Staff retention and development
  - Freedom from vendor lock-in
  - Valuable business arguments, if difficult to measure

➢ **Open Source is the future—embrace it!**

**Good function / Vendor independence / Lower cost**

# Questions?

Phil Smith III

LEVANTA

703.476.4511

psmith@levanta.com

www.levanta.com