Open Source:
Making A Business Case

Phil Smith III
Linuxcare, Inc.
SHARE 99
August 2002
Session 9320

Copyright Information

Disclaimer

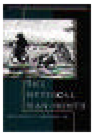
No animals were injured in creating this presentation. Void where prohibited by law. All rights reserved. Your mileage may vary. Available while quantities last. Used with permission. Presentation was current at time of printing. Abandon hope all ye who enter here. For research purposes only. Any resemblance 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

Required Reading

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



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 the ground up
  - Developers hate reinventing the wheel

The Shortest Answer

- Microsoft
  - …or Sun
  - …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
  - Correct behavior (at least somewhat) defined
- Publicly supported
  - Lots of others to give help with problems
  - "Community" development aspect is appealing
- Enables use of 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 people to fix others’ problems”
- 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
  - 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
  - Those 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
- RAD
  - Building on an Open Source application saves big development $$$ ***and time***
- Openness promotes exploitation
  - Add-ons, ideas, enhancements (cf. id & Doom)

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 our stuff away”
- “Proven” failure of model
  - VA Linux, Netscape, et al. – “They’re all dead”

What About Bill?

Microsoft

Microsoft Is Not Evil

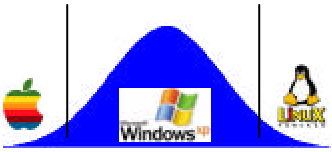
- Redmond is a ***business***
  - Has provided excellent return to stockholders
- Products are carefully designed and developed
  - Tens of millions of users love them!
- Consumers' willingness to put up with BSODs, etc. are our fault, ***not*** Bill's
- Apparently the value of Windows is greater (for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

Microsoft Isn't Perfect

- *De facto* monopoly position reduces quality
  - Little or no corporate incentive to produce products of truly ***excellent*** quality
- Internet service distribution (Windows Update) greatly reduces service costs
  - Alas, it may further dampen initial product quality

The Reality

- Windows is "good enough" for most folks
- A classic bell curve distribution:



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

Microsoft May Be Doomed

- Microsoft's business model depends on customers upgrading to newer releases
  - Open Source applications threaten their ability to cram upgrades down consumers' throats
- PCs have passed the point where newest, fastest 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 A. Hun, G. Khan, A. Hitler…
- 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!
  - See www.opensource.org/halloween/

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
  - You buy them to run applications
- Operating Systems are a dead end without new applications
  - At some point you’re ***forced*** to upgrade
- But “it works well enough” is compelling!
  - DOS, Windows 3.1/95, old Macs in daily use

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!

Why Was That Good?

- Detailed staff knowledge of internal applications was considered competitive advantage
  - Obviously not true of unmodified Open Source
  - But self-destiny (fixability) still big advantage
- Intangible but real:
  - Staff “big picture” vision enabled avoiding some stupidity (cf. CRM disasters…)

How Is Open Source Different?

- Brooks’ Law seems not to apply:
  - Complexity and bugs do ***not*** rise with the square of the number of programmers involved!
- “Hacker culture” fundamentally different from traditional development culture
  - “Gift culture” makes knowledge-sharing valuable
  - Contrast with “proprietary advantage” theology

The Car 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 Emotional Appeal

- Emotional arguments of “I hate Microsoft” (or Oracle, or…) are ***not*** business cases!
  - Many fail to recognize emotional involvement
- Consider the outrage over Microsoft “stealing” the GUI from Apple
  - Few are outraged that KDE and Gnome “stole” the GUI from Windows (or even Apple)

Cars Again…and Bourgeoisie

- Stephenson also compares the emotional allure of Linux to that of the MGB:
  - Difficult to maintain
  - Requires technical knowledge just to live with
  - “Separates the men from the boys”
  - Windows is “tacky”
  - Microsoft is evil simply because they are large
  - Enables “purity of essence” by avoiding Windows
- This is a fundamentally elitist attitude! (But is that bad?)

When Open Source?

- Raymond suggests that appropriateness depends on program use:
  - **Most appropriate:**
    Infrastructure (e.g., Samba, DNS)
  - **Less appropriate:**
    Middleware (e.g., databases)
  - **Least appropriate:**
    Applications (e.g., word processors)

The Distinction?

- **Infrastructure:**
  Commoditized, strong standards
- **Middleware:**
  Semi-commoditized, weaker standards
- **Applications:**
  Not commoditized, few or no standards
- As software evolves, it moves toward the “more appropriate” end of the spectrum

Something “Wants” to be Free?

- Stephenson further suggests that Operating Systems are “destined to be free”
- Remember why we have computers: applications!
- If sufficiently decent applications are available for free OSes, they will dominate
  - Microsoft will be forced to acknowledge this eventually
  - Netscape talked about commoditizing the OS, which is what caused Microsoft to take aim at them

When Not Open Source?

- **Integration issues:**
  Vendor apps typically better integrated
  - Although “Integration” often means “We put it all on one CD”
- **Customization issues:**
  Some types of applications “always” require significant custom work
  - E.g., CRM implementations

When Not Open Source?

- **Standing investment:**
  Existing, paid for product licenses
  - Consider switching at next upgrade cycle
- **Training and conversion:**
  Costs can be surprisingly high
  - Business units 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 vendor application 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?
  - “Closed-source when Open Source choices exist will be…grounds for shareholder lawsuit”

Some More Issues

- Open Source typically more secure
  - Reading source exposes weaknesses
  - Availability of fixes often measured in minutes
- Vendors can provide support “guarantees”
  - Can they live up to them?
  - If they don’t, what remedies do you have?
- Mission-critical applications require serious support—no question

Still More Issues

- Depending on platform, bugfixes for OSes 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

Making The Decision

The Real Argument(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)

So 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: control spending on labor, facilities, hardware, software
- Open Source can (sometimes) help with all of these
  - **Labor:** many Open Source apps very mature
  - **Facilities:** server consolidation can save big
  - **Hardware:** server consolidation again
  - **Software:** the most obvious opportunity

Summary

Summary

- Primary Open Source drivers are 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 argument, if difficult to measure

➢ **Open Source is the future—get used to it**

Contact Info

Phil Smith III

LINUXCARE

703.568.6662
psmith@linuxcare.com
www.linuxcare.com