Competition and Sustainability: Two Sides of the Same Coin

Michael Tiemann
Vice President, Open Source Affairs

Origin of Innovation in America

"When a private individual mediates an undertaking, however directly connected it may be with the welfare of society, he never thinks of soliciting the cooperation of the Government, but he publishes his plan, offers to execute it himself, courts the assistance of other individuals, and struggles manfully against all obstacles. Undoubtedly he is often less successful that the State might have been in his position; but in the end **the sum of these private undertakings far exceeds all that the Government could have done**."

-- Alexis de Tocqueville, Democracy in America

n.b. It is not the choice between private monopoly-like approaches that is better than a government-originated monopoly, but the competition, choice, and most importantly **the sum total of multiple, interoperable, and cumulative results**.

redhat

In 1987...

- Compiler ports cost $1.5M-$5M and took 2-3 years to deliver
- The National 32032 chip was marketed as a Motorola 68K-killer
  - 32-bit vs. 16-bit architecture
  - “orthogonal” (VAX-like) instruction set
  - True 1 MIPS performance (1 full VAX Unit of Performance)
- When delivered, 32032 was only 0.75 MIPS/VUPS
- The day that GCC was released as free software (supporting VAX and m68k), I decided to attempt a port to the 32032
  - New port + 20% better performance after two weeks
  - 40% better performance after four weeks
  - GCC delivered 1.4*.75 = 1.05 MIPS for free, but National would not abandon their multi-million dollar investment in failed technology
- The National 32032 died; National exited microprocessors

In 1987...

- Los Alamos invested $100M in Sun Microsystems workstations to create a “virtual” atomic bomb
- Visiting our lab, I issued them a challenge: tell me their #1 most important computing routine, then before they left that afternoon, I'd deliver better performance with a free compiler than Sun ever did
- In four hours, with no documentation, I delivered 10% better performance on their most critical routine (and many others, too)
- 10% of $100M is $10M of excess value created in four hours
- I was invited to the lab to meet the Director, who, after avoiding me all day, told me “we have a way of doing things around here, and we're not going to change that just because of what you have done. How does that make you feel?”

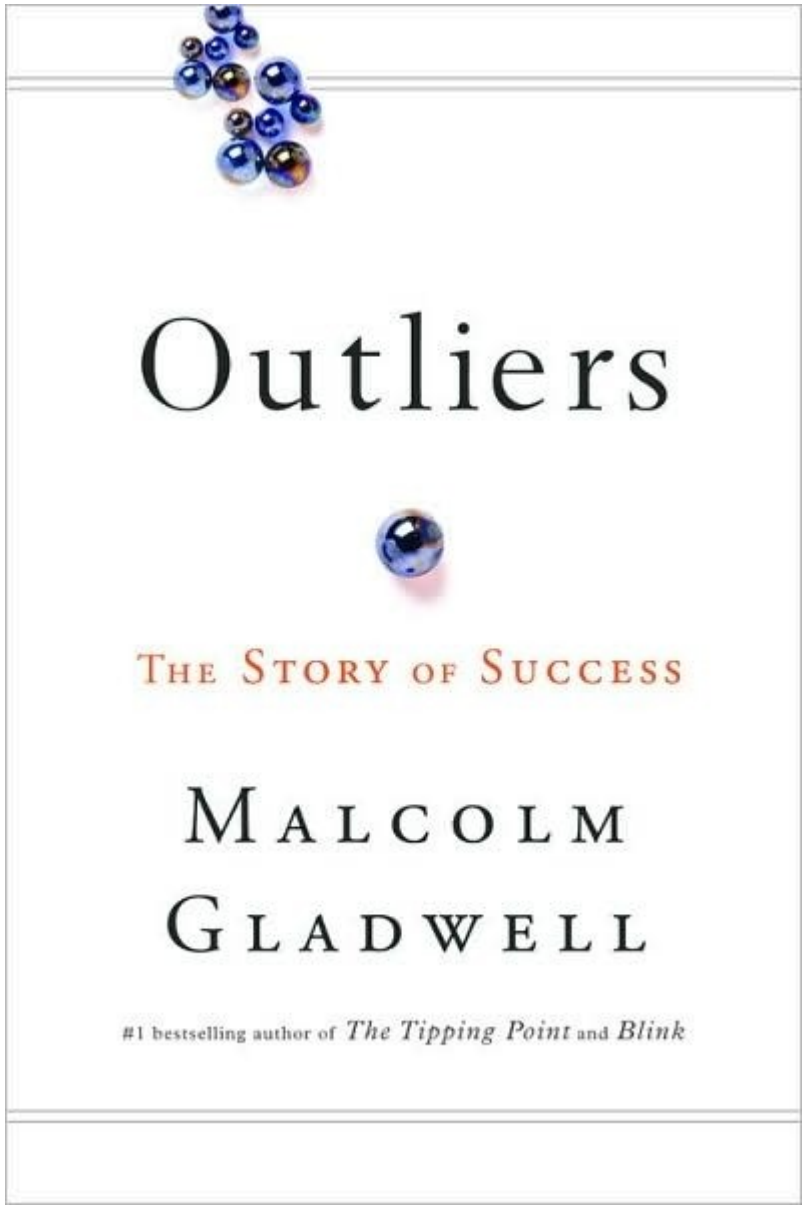
In 1987...

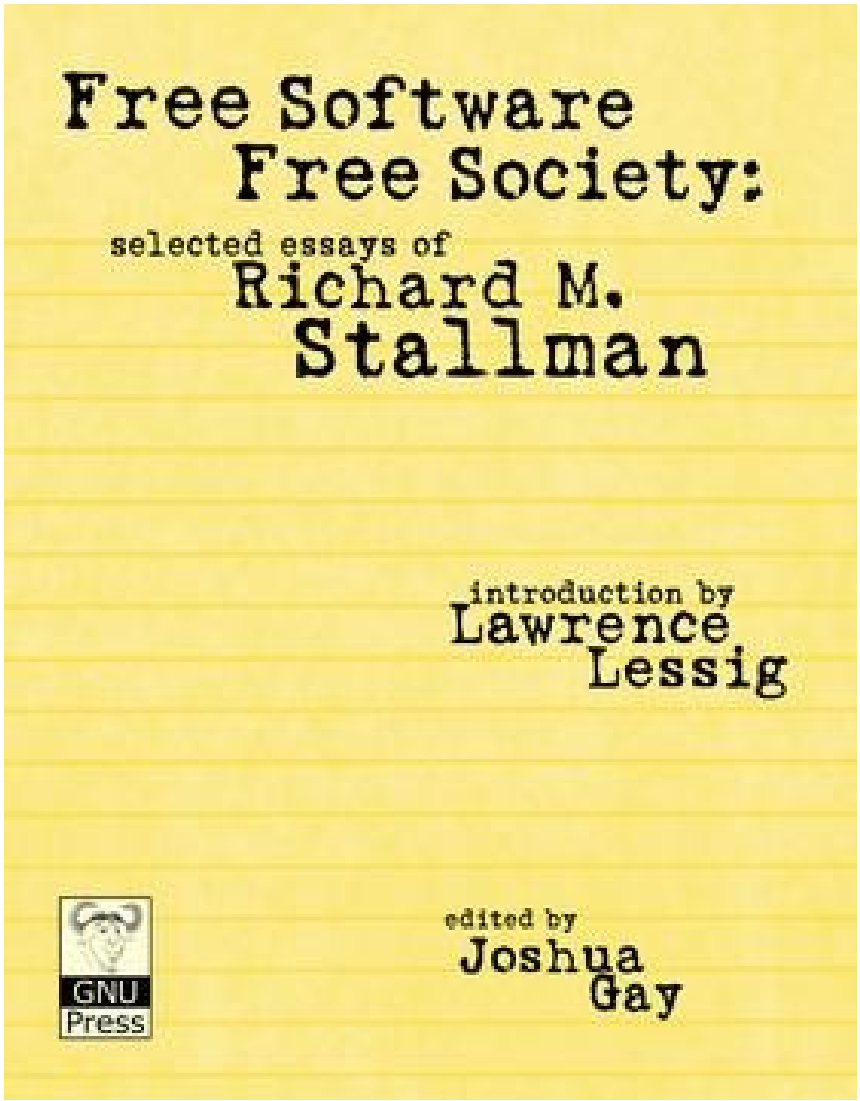
- A month after the Los Alamos visit, I received a mysterious package by Federal Express – the architecture reference manual for the SPARC microprocessor
- Three days later, I finished the port, generating competitive performance to the Sun compiler, but I didn't know whom to tell
- It took me one more day to tune the compiler to better performance than Sun's own compiler, and to deduce the identity of the mysterious correspondent
- When I called to tell him of my exploits, he offered me a job at Sun
- I told him I'd come in a year, after delivering on my promises to DARPA (which I did)

In 1987...

- The C++ programming language was rising rapidly (and paving the way for Java). It's creator, Bjarne Stroustrup, would later be recognized:
  - 1990: Top 12 young scientists, Fortune Magazine
  - 1993: ACM Fellow, and Admiral Grace Hopper Award winner
  - 1995: 20 Most Influential people [computer industry in] past 20 years
  - 1996: AT&T Fellow
  - 2004: Member, National Academy of Engineering
  - 2005: IEEE Fellow
  - Did not write a native code C++ compiler because that was “too hard”
- I released the first native code C++ compiler December, 1987
- 5 years later, 30+ people at Bell Labs abandoned their effort
- Today, even Apple uses GNU C++

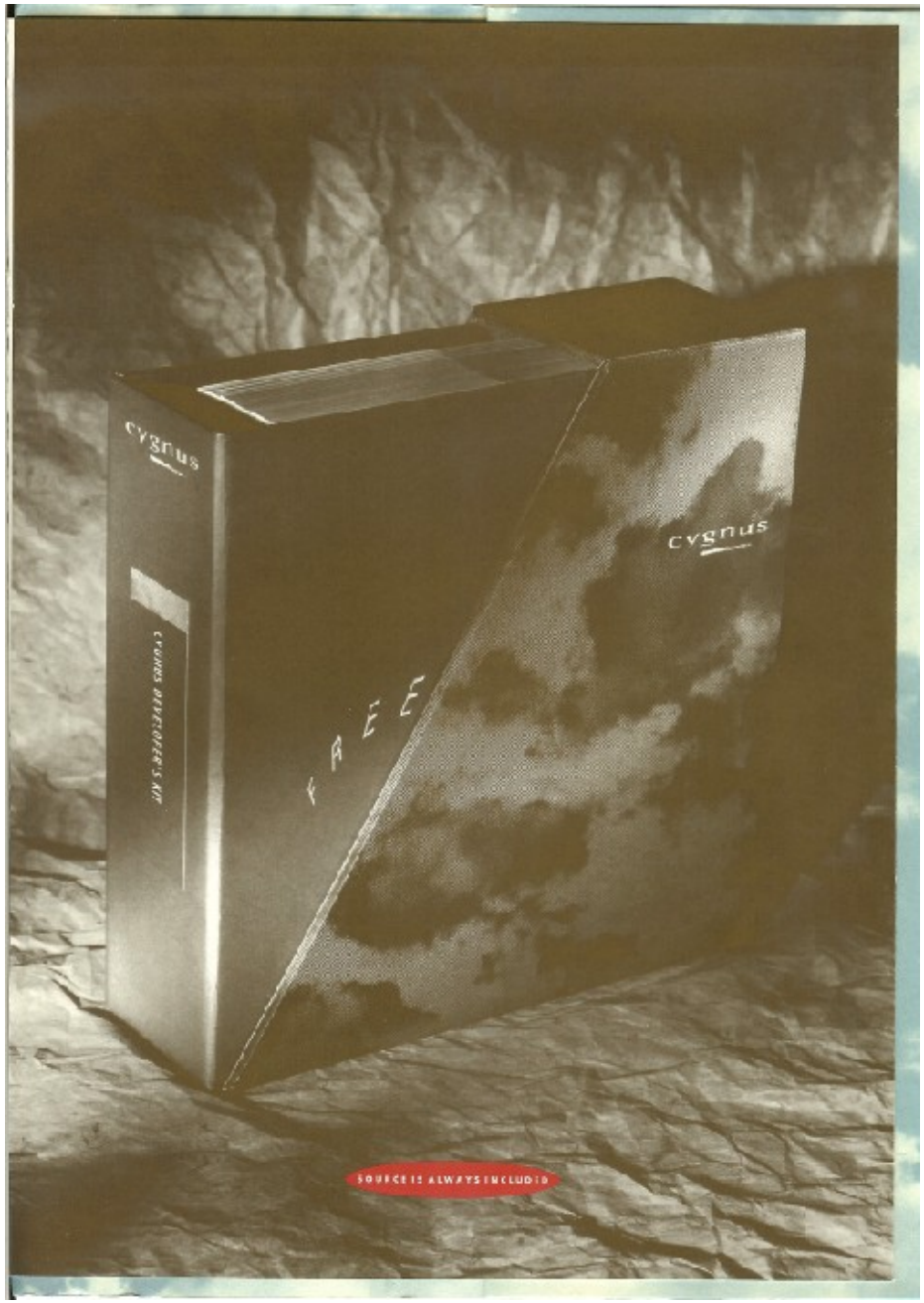
Was this about me, or about free software?





redhat

I believed it was about freedom...



...and that the success of a company based on free software could fundamentally transform the industry.

redhat

Along the way, we invented:

- One of the first commercial uses of the Internet (cygnus.com)
- The ISP (the little garden)
- The software subscription model
  - Leveraged, Progressive, and Vintage Support
- The first Free Software magazine (the Free Software Report)
- The first working POSIX environment for Windows (cygwin)
- A dual-licensing model for free and proprietary software (cygwin)
- A fully generic software build system (autoconf, automake)
- Free software tools for: regression testing, bug tracking, library management
- Public-facing, internet-enabled Christmas Tree
- And of course...the first Open Source company

Standards and Control

"The decision to make the Web an open system was necessary for it to be universal. **You can't propose that something be a universal space and at the same time keep control of it.**"

-- Tim Berners-Lee, Creator of the World Wide Web

See http://www.w3.org/People/Berners-Lee/FAQ.html#What2

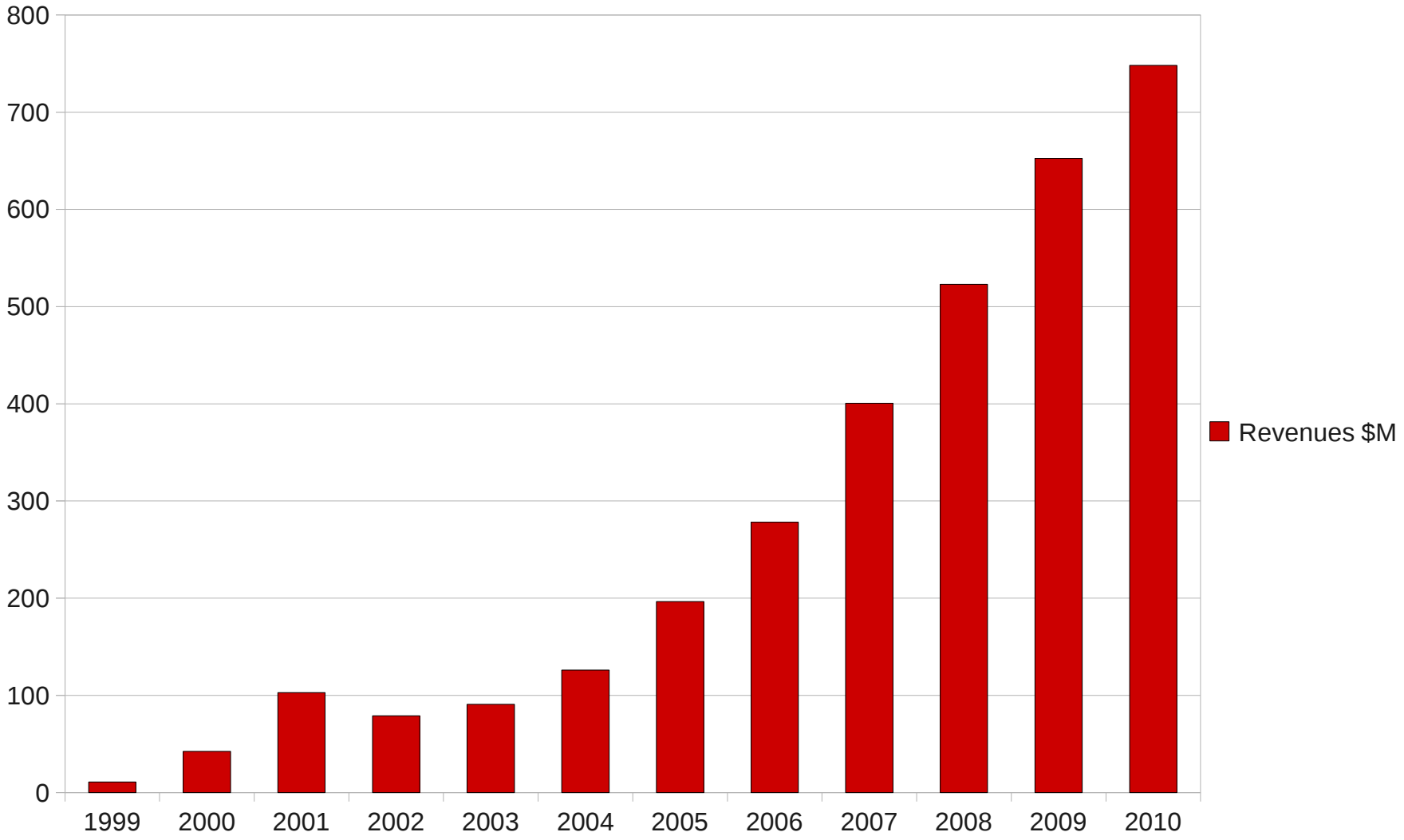
redhat

redhat

Vital Statistics

- Headquarters in Raleigh, NC
- Founded in 1993
- IPO, 1999
- Acquired Cygnus 2000
- S&P 500 (NYSE: RHT)
- FY10 revenues: $748 million
- 3,200+ employees in 28+ countries
- Cash and investments: $761 million (virtually debt-free)

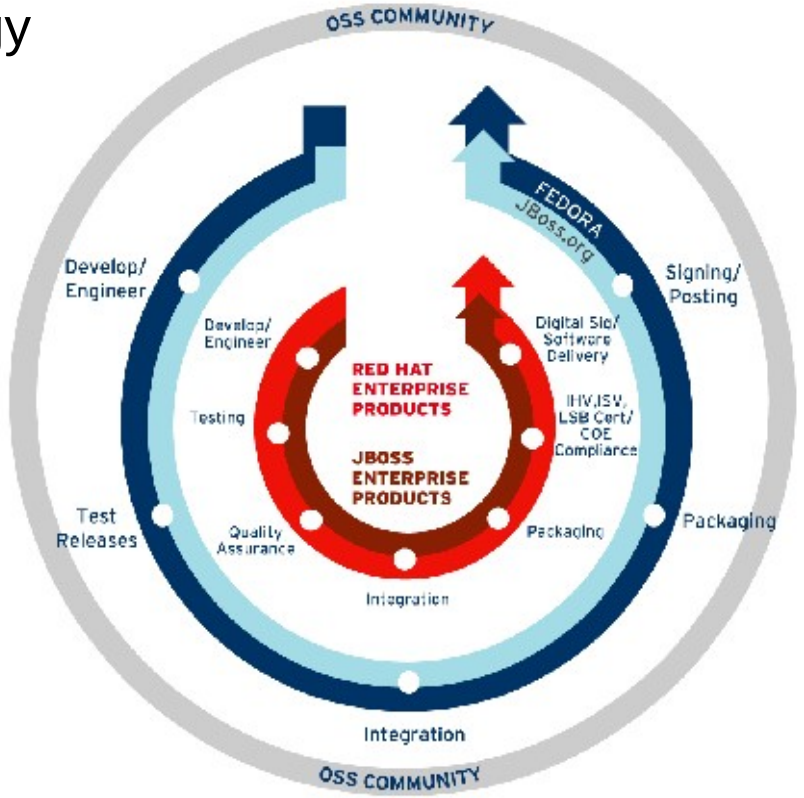
Red Hat Revenues (1999-2010)



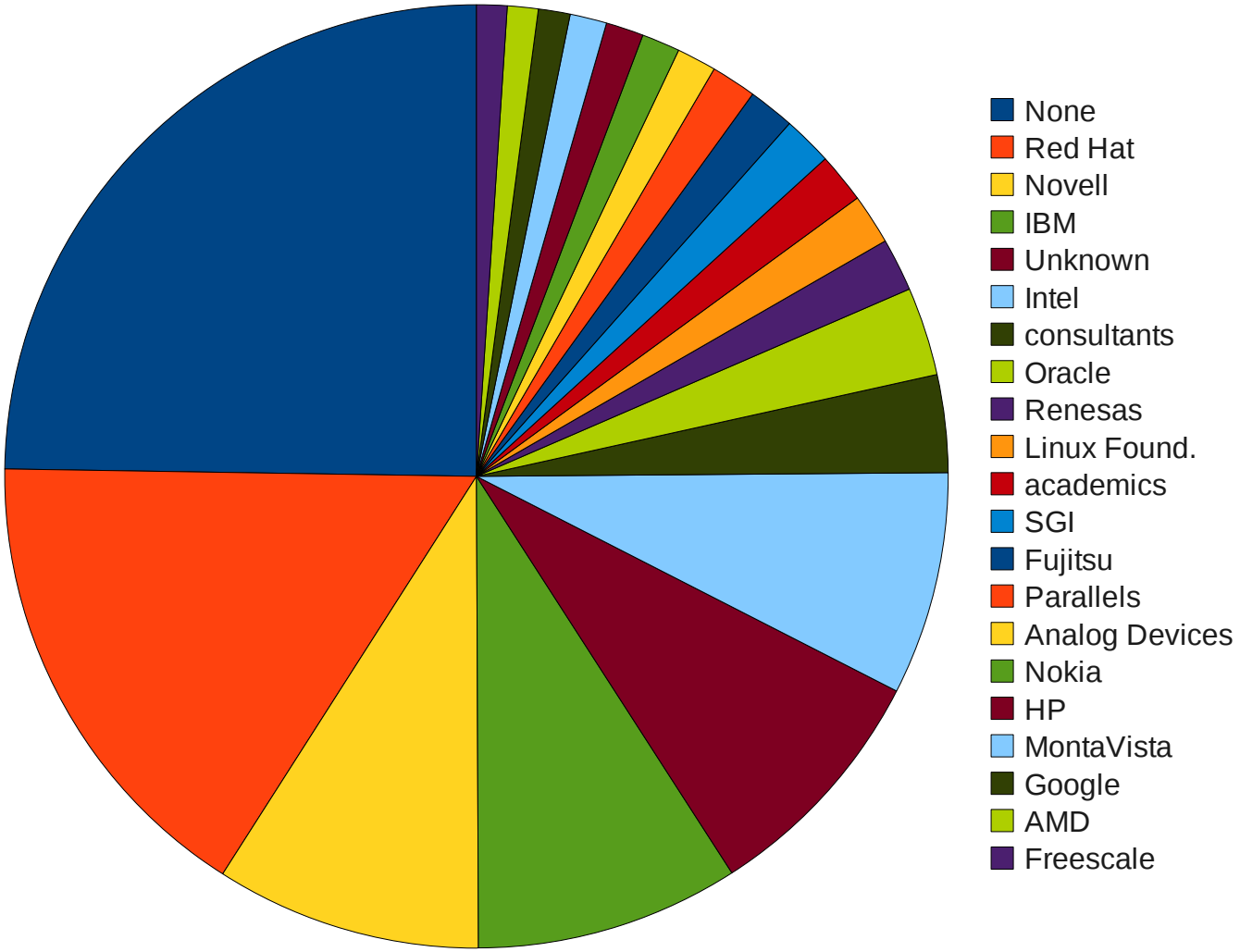
redhat

Making a Project Into a Supported Product

- Collaboration with partners and open source contributors to develop technology
- Deliver complete distributions in two stages for two audiences
  - First stage
    - Fedora and JBoss.org– development vehicles
    - Approximately twice/year
    - Unsupported
    - Fast moving, latest technology
  - Second stage
    - Red Hat Enterprise Linux/JBoss Suite
    - Approximately every 18 months
    - Supported and certified
    - Stable, mature, commercially focused technologies



Major Contributors to Linux



Source: Linux Foundation 2010

redhat

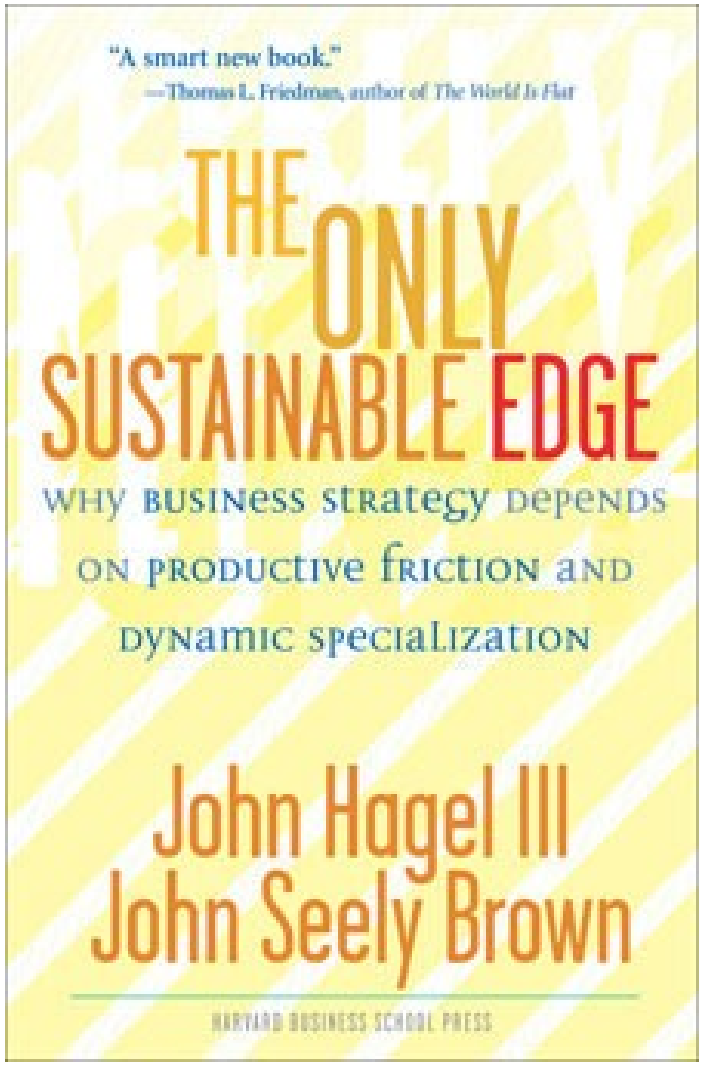
Was this about Red Hat, or about Community?

TOP 10 SOFTWARE VENDORS IN 2010

RANK 2010	RANK 2009	RANK 2008	RANK 2007	RANK 2006	RANK 2005	RANK 2004	VENDOR	AVERAGE OF ALL RATINGS 2010 (%)	VALUE (%)	RELIABILITY (%)	PREFER TO STAY WITH VENDOR (% YES)
1	3	-	-	-	-	-	WebEx (Cisco)	78	78	81	95
2	1	2	1	1	1	1	Red Hat	74	74	79	91
3	1	1	2	-	-	-	Google	71	71	74	94
4	4	3	3	2	7	6	Citrix	71	71	73	91
5	7	6	-	-	-	-	Salesforce.com	66	66	69	84
6	6	7	6	4	31	36	Microsoft	66	66	69	90
7	5	4	4	-	-	-	Adobe	65	65	68	93
8	9	8	8	6	39	28	Oracle (including PeopleSoft)	61	61	64	83
9	10	9	5	5	26	24	SAP	60	60	62	79
10	8	5	7	3	22	17	Novell	55	55	56	74

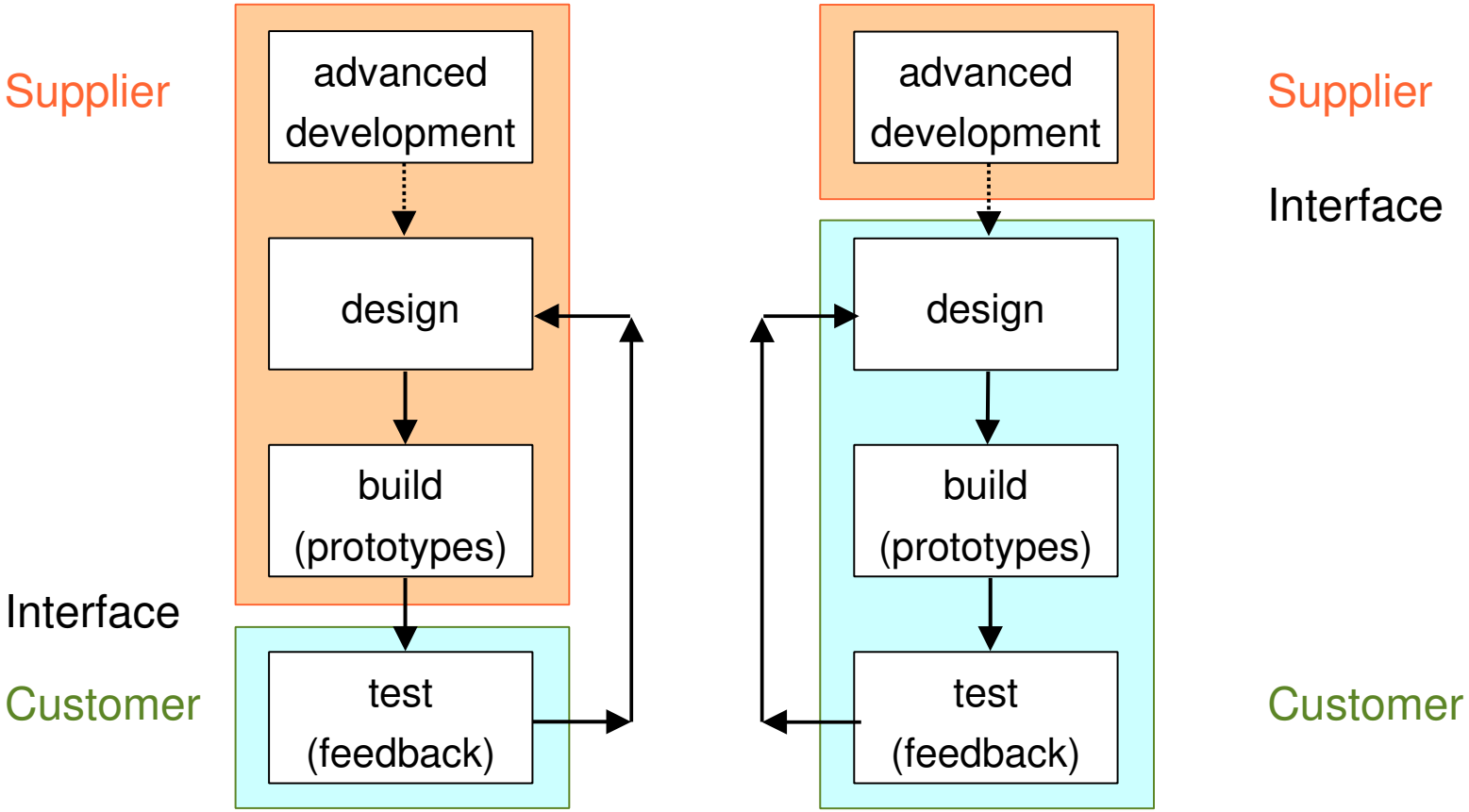
Source: *CIO Insight Vendor Value 2010*

I began to think it was about Community... and liberal distribution



redhat

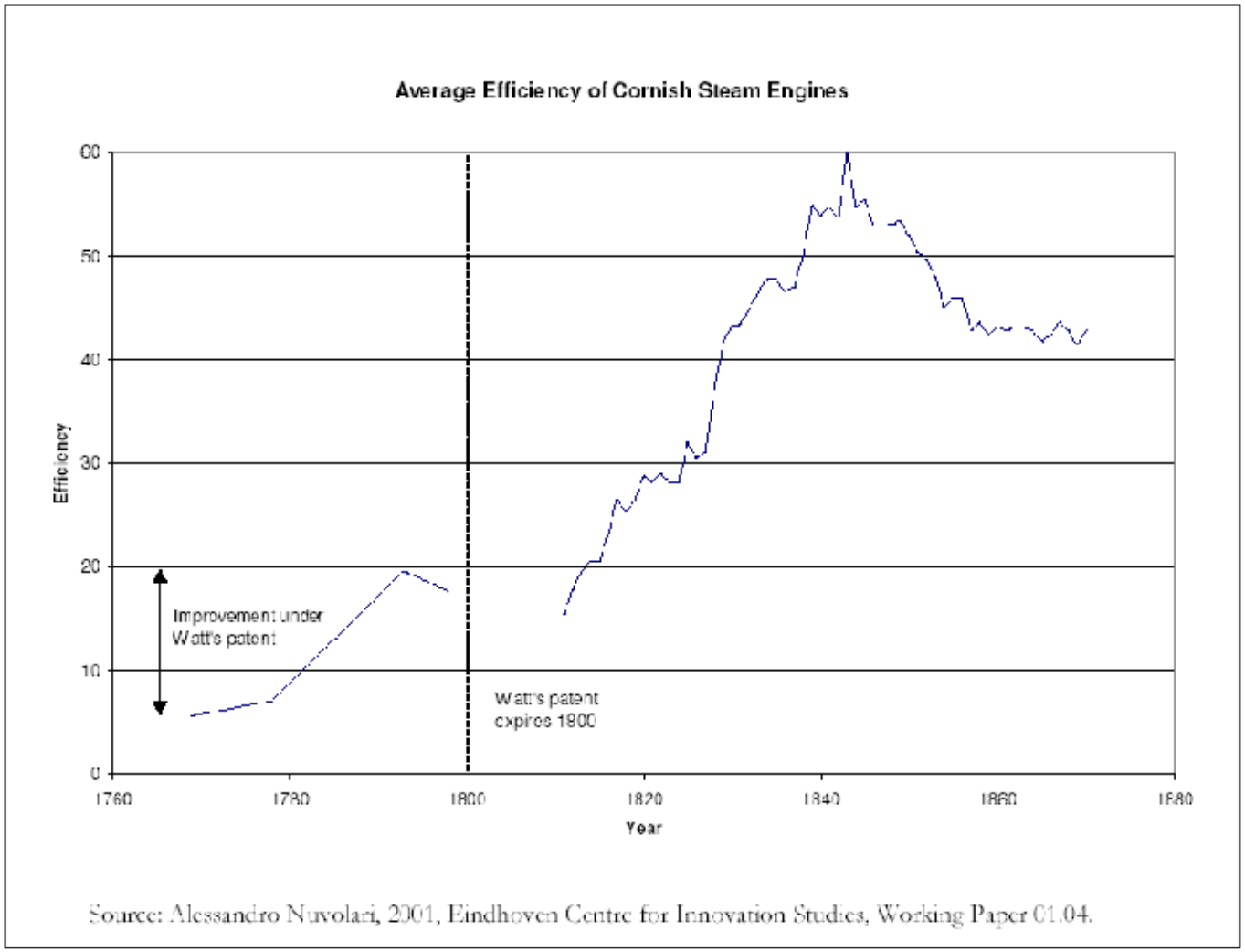
From #1 in industry to a whole new industry



Thomke, Stefan and Eric von Hippel (2002) “Customers as Innovators: A New Way to Create Value“ Harvard Business Review, Vol 80 No. 4 April pp 74-81.

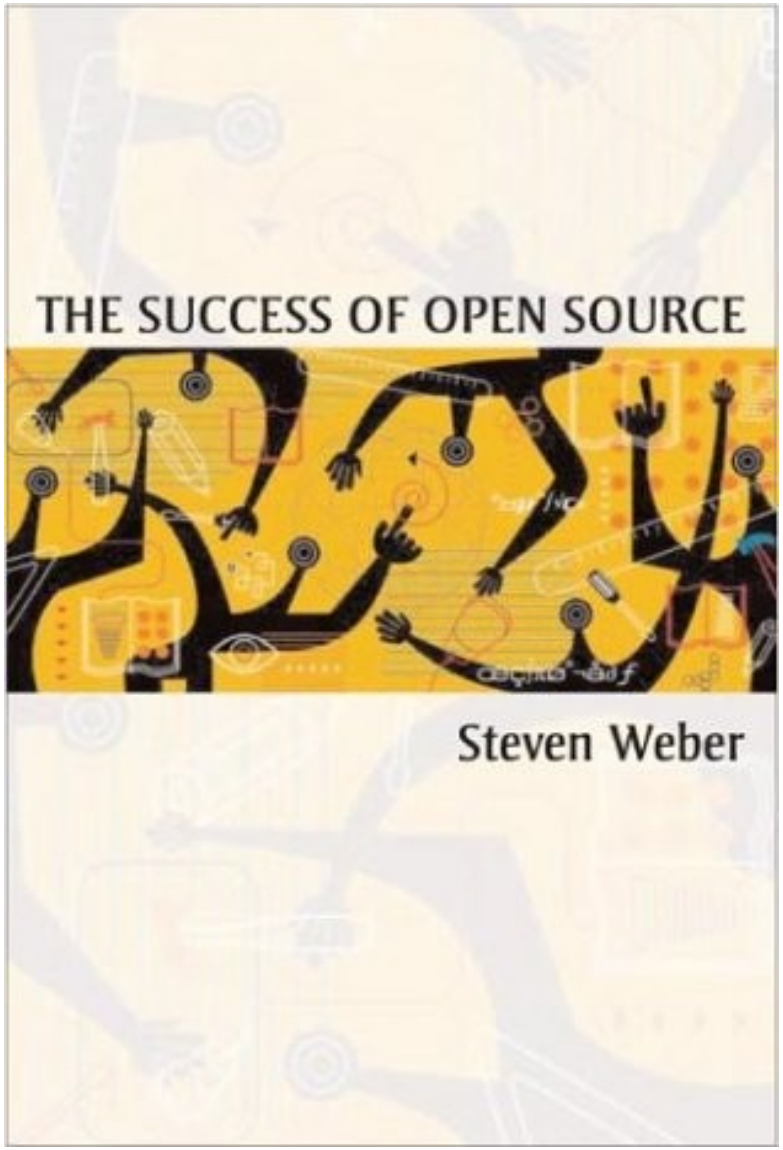
redhat

Necessity is the mother of invention...



Source: Alessandro Nuvolari, 2001, Eindhoven Centre for Innovation Studies, Working Paper 01.04.

redhat



"The conventional notion of property is the right to exclude. Property in open source is configured fundamentally around the right to distribute, not the right to exclude."

Prof. Steven Weber
Director of the Institute of International Studies
UC Berkeley

redhat

Moore's Cannibal Principle

"The whole point of integrated circuits is to absorb the functions
of what previously were discrete electronic components,
to incorporate them in a single new chip, and then to
give them back for free,
or at least for a lot less money than what they cost as individual parts.
Thus, semiconductor technology eats everything,
and people who oppose it get trampled."

Source: Gordon Moore (Intel Chairman) quoted in Brent Schlender,
Why Andy Grove Can't Stop *Fortune*, July 10, 1995, p. 91

But what about Sustainability?

Designing for Difficulty: A Long Way To Go

"Even in 1909, the fundamental limitations of [the Wright Brothers'] design are evident. Much the way a bicycle cannot maintain its balance unless it is moving, the Wrights have purposefully designed their planes to be inherently unstable, believing, mistakenly, that this is an essential factor to control in the air." From Unlocking the Sky by Seth Shulman

"Bad Software" is software that was intentionally designed to hamper or completely thwart rivals, even when such manoeuvres hurt not only the software itself, but the customers of that software; See Breaking Windows by David Bank

- 2001: The Standish Group Estimated $78B/year wasted on "Bad Software"
- 2002: NIST Estimated $60B/year lost in US alone due to "software bugs"
- 2002: Net profits of Fortune 500 is approximately $68B
- 2003: US Federal IT budget set at $59B
  - History suggests 80% will be wasted, not deployed
- 2003: Cost of Worms and Viruses alone range $17B-$55B

The True Cost of “Bad Software”

[However], there [has been] no Moore's law for software.
While computing power falls rapidly in price, software that can make use of that computing power becomes more complicated, sometimes more expensive and less reliable, and almost always more difficult to configure and maintain.
Yet it is software that constitutes the fundamental rules for information processing, and thus for an information economy and an information society.
Massive processing power connected by ever-increasing bandwidth is a skeletal infrastructure. Software determines how information is manipulated, where it flows, to whom and for what reasons.

--United Nations Conference on Trade and Development (UNCTAD) 2003 p.95

The State of ICT and Software, 2010

"We can't solve problems by using the same kind of thinking we used when we created them." – A. Einstein

- We have quite some problems in IT:
  - $1.3T USD Enterprise IT spending
  - 18% IT projects abandoned before production
  - 55% "challenged" (late, lacking, broken)
  - $500B USD is ***wasted*** due to "bad software"
  - $3.5T USD anticipated value ***not delivered***

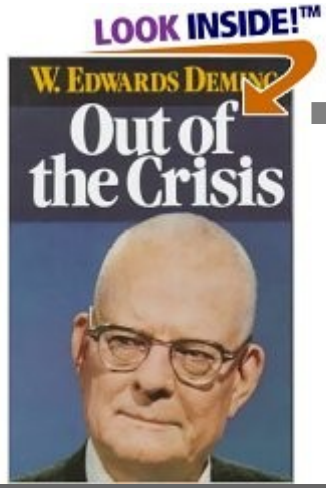
- Proprietary software model is not sustainable
- Open Source is a new approach that can radically improve both IT and the businesses that use it

Reference: http://opensource.com/business/10/6/integral-innovation

redhat

Out of the Crisis – Deming (1982)

- Create constancy of purpose
- Adopt new philosophy/change
- Build quality in the first place
- Build relationships around loyalty and trust, not price
- Improve product and service constantly and forever
- Improve people w/training
- Replace supervision with leadership

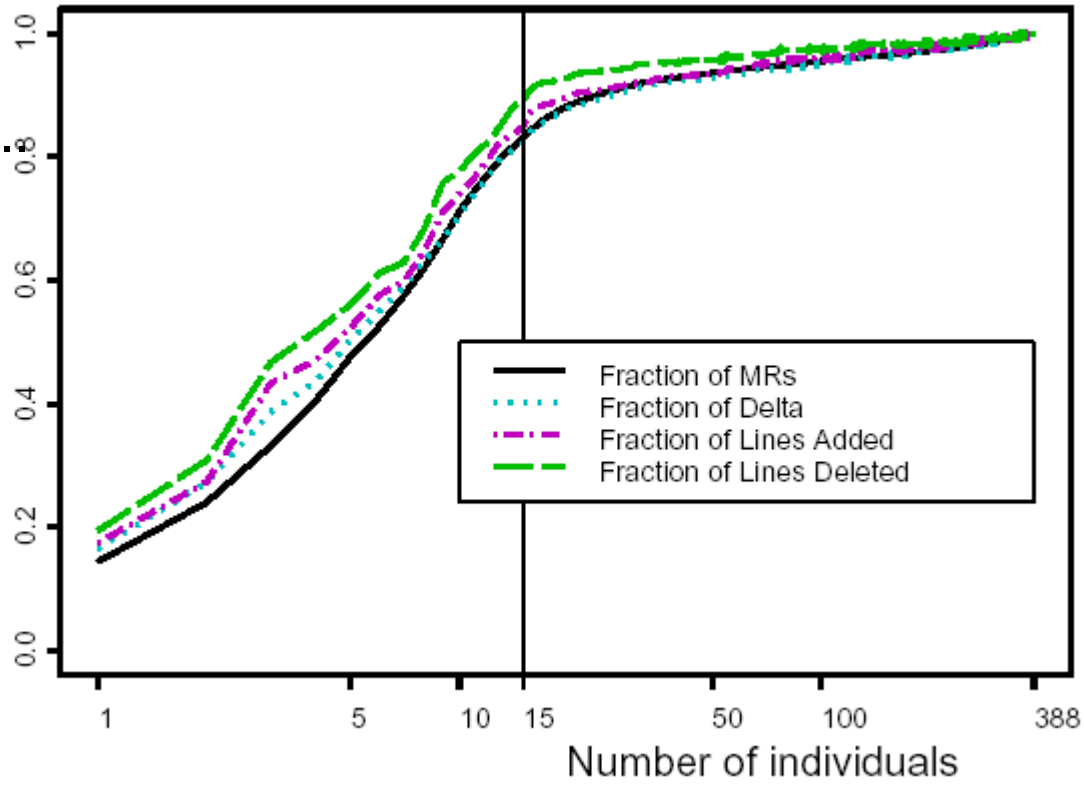


- Drive out fear so that everybody can participate
- Break down barriers between departments; work as team
- Eliminate adversarial relationships
- Replace quotas, MBO, etc. with leadership
- Restore pride of workmanship by rewarding quality, not numbers
- Strongly support programs for self-improvement
- Transformation is everybody's job

redhat

The Long Tail of Open Source

- OSS achieved first article sooner...
  - With fewer bugs...
  - That were fixed sooner...
- The trend continues...
  - Xen Virtualization
  - SE Linux
  - GRASS/R/PostgreSQL
  - MySQL
  - JBoss ecosystem
  - Eclipse
  - Blender, Inkscape, GIMP, Ardour, Audacity, etc.



Source: http://opensource.mit.edu/papers/mockusapache.pdf

redhat

Observed results—Quality

- Typical proprietary software would has 20-30 defects per 1,000 Source Lines of Code (SLOC)
  - Or 114,000 to 171,000 defects per 5.7 MLOC
- 2004: Coverity finds 985 defects in Linux kernel
  - 627 defects found in critical parts of the kernel
  - **100% of “serious” defects fixed in 6 months**

    http://www.eweek.com/c/a/Linux-and-Open-Source/Linux-Kernel-Review-Shows-Far-Fewer-Flaws/
- 2005: Defect density down from 0.17 to 0.16
  - Defect density declined 2.2%
  - Code size increased 4.7%

    http://www.internetnews.com/dev-news/article.php/3524911

Observed results—Quality

- 2006: Average of 32 OSS programs is 0.434 per KLOC
  - Perl @ 0.186 defects per KLOC
  - GCC @ 0.202 per KLOC
  - Python @ 0.372 per KLOC
  - http://www.internetnews.com/stats/article.php/3589361
- No correlation between size and defect density
  - No “black holes” in terms of quality
- LAMP defect density is currently 0.29 per KLOC
  - PHP worst @ 0.474

Observed results—Quality

- 2008: Average of 250 OSS programs is 0.33 per KLOC
  - PHP was “perfect” with zero detectable defects
  - 10 other projects also “perfect”

    http://scan.coverity.com/report/Coverity_White_Paper-Scan_Open_Source_Report_2008.pdf
- 2009: Average of 280 OSS programs is 0.25 per KLOC
  - 36 projects now “perfect”

    http://scan.coverity.com/report/Coverity_White_Paper-Scan_Open_Source_Report_2009.pdf
- 2010: Accenture survey finds Quality is #1 reason why enterprise customers choose OSS (Cost is #5)

  http://opensource.com/business/10/6/integral-innovation

“Whatever you do will be insignificant. But it is very important that you do it!” – M. Gandhi

Protection v. Innovation

Developer 1 \ Developer 2	Don't Work	Work
Don't Work	0,0	v, v-c
Work	v-c, v	v-c, v-c

v: value to developer
c: cost to developer

Developer 1 \ Developer 2	Don't Work	Work on A	Work on B
Don't Work	0,0	.5v, .5(v-c)	.5v, .5(v-c)
Work on A	.5(v-c), .5v	.5(v-c), .5(v-c)	v-.5c, v-.5c
Work on B	.5(v-c), .5v	v-.5c, v-.5c	.5(v-c), .5(v-c)

- Game theory predicts: more modules and more option value leads to more developers (http://www.people.hbs.edu/cbaldwin/DR2/BaldwinArchPartAll.pdf)
- More than 2M OSS developers working on more than 1B SLOC proves game theory is good theory (http://www.springerlink.com/content/q551lwg63762n24l/)

Upton's Path-based Model

	Installation Based	Path Based
Role of IT	Supportive/Peripheral to Operation	Integral part of Operation
Project Size and Number	Large, few, infrequent	Small, many, frequent
Development Approach	Build, then install	Prototype and evolve
Delivery of Value	When a project is complete	On-going
Source of Technology/ Software	Heavy use of proprietary interconnection code, proprietary standards	Standards in common use
Primary Funct'l Concerns	Control, efficiency, accommodating all requirements at once	Integration, interconnection, flexibility, progressive delivery of req's
Locus of Technical Control	Vendor/IT group	Operation itself
Experimentation	Limited	Frequent opportunities

See http://www.people.hbs.edu/dupton/papers/pathbased-it/PATH.PDF

Revised May 27, 1997

redhat

Open Source Security—NSA's SE Linux Project

- Built on 10 years of NSA's OS security research
- Application of NSA's Flask security architecture
- Cleanly separates policy from enforcement using well-defined policy interfaces
- Allows users to express policies naturally and supports changes
- Fine-grained controls over kernel services
- Transparent to applications and users
- Role-Based Access Control, Type Enforcement
- *Initially rejected as “impossible”*

Sustainable Security

- SE Linux succeeded with “The Open Source Way”
  - 14 initial policies supported in Fedora Core 3
  - 80+ policies supported in Fedora Core 4
  - User-loadable policy management in Fedora Core 5
  - Now thousands of protected apps, services, etc.
- Five years of RHEL4: Zero critical kernel exploits
- Three years of RHEL5: Zero critical kernel exploits
- Cloud computing and virtualization make security *more* important, not less!
- Breaking down barriers helped build better barriers!

Thank you!