15.351 Managing Innovation and Entrepreneurship
Spring 2008

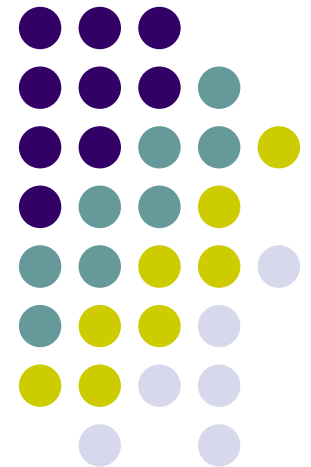# 15. 351 Managing Innovation & Entrepreneurship

Fiona Murray

MIT Sloan School of Management

Spring 2008

Class Thirteen

Linux & IBM 1999

- Not so secret 2-day summit in September 1999 with 260 IBM managers
- Linux Technology Center opened with US$5M – 25FTEs
- Delicate, slow and quiet approach
  - Listen and understand
  - Don't overwhelm
  - Only the "Best" programmers
- Two reports to the Corporate Technical Committee
- Establish Open Source Steering Committee

Linux & IBM 2000-2002

- January 2000 – embrace Linux, all servers Linux enabled

- All three server O/S development efforts (Aix, Monterey and Linux) report to one person

- November 2001 – IBM to invest US$1BN in Linux

- 2002 – Open Source Development Labs established (employs Linus Torvalds)

Linux & IBM 2003-2005

- 600 IBM programmers work on Linux ($120 M)
- More than 80% of IBM submissions are accepted into the 2.5 kernel
- IBM "donates IP" (over 500 patents) to a variety of open source communities

- Did they make $$$? hard to tell…

Lessons Learned: How can you tell if a project is likely to succeed?

Business value & open source [project] vision align
- Success Stories: Linux, Apache
- Progressing: Eclipse, Globus

Critical Factors for a successful Open Source Community
- Business perspective
  - ✓ Must provide "business value" to end customers
  - ✓ Active sponsorship and marketing by 1 or more significant IT vendors
  - ✓ Solid overall vendor support
  - ✓ Critical skills available and actively engaged
  - ✓ Long-term plans for improvements - plans, design, development, test, documentation

- Developer perspective
  - ✓ Passionate interest in developing and enhancing code
  - ✓ Significant overlap between set of users and developers - common wavelength
  - ✓ Personal rewards - able to build status and recognition in developer community
  - ✓ Diverse, interactive community
  - ✓ Strong overall project/code leadership (e.g. Linus Torvalds -Linux)

Adapted from: IBM OpenSource Presentation

Lessons Learned: When can you tell a project is likely to fail?

Heavy on technology – light on business input/understanding
- Lack of continued customer interaction/business insights

Poorly - or no - thought through developer community evolution & ecosystem
- Failure: Open source it….they will come
- Failure: Build it, productize it - ot a level playing field, etc..

Critical Factors for a successful Open Source Community
- Business perspective
  - Provides "business value" to end customers
  - No Active sponsorship and marketing by 1 or more significant IT vendors
  - No Solid overall vendor support
  - No Critical skills available and actively engaged
  - Long-term plans for improvements - plans, design, development, test, documentation

- Developer perspective
  - Passing interest… No continued interest in developing and enhancing code
  - Significant overlap between set of users and developers
  - Personal rewards - able to build status and recognition in developer community
  - No Diverse, interactive community
  - No Strong overall project/code leadership (e.g. Linus Torvalds -Linux)

Adapted from: IBM OpenSource Presentation

Key issues managing talented technical professionals

- **Motivation**
  - \>> complex than simply money (salary or options)
  - Challenging fun projects…
  - Sometimes money can hinder
  - Recognition key – peer recognition critical (science, open source)
  - Multiple motivation – pragmatists, idealists, enthusiasts, skill builders, careerists

- Consider how motivation & rewards are connected to **monitoring & measurement**

    AND

- How the work of these individuals is "**controlled**"/directed & incorporated into the technology S-curve

Mako Hill

- Publication of "Cathedral & bazaar" (Eric Raymond)
- Initiated Netscape openness…
  - => Netscape releasing browser code in 1998 – made wrong assumptions - Netscape had "freed" a piece of software and thought people would want to work on this….but they didn't..code was in accessible…"I went to write support for window pop-up feature but couldn't figure out where to add the code…I couldn't do it…not familiar with the code base…no documentation…it was the job of those in Netscape but unworkable in the community….so well meaning users couldn't….also hard to build….didn't open their build computers…..someone would go onto a mailing list and give their idea and the Netscape people would "steal" their ideas….so even contributors who wanted to contribute were not able to…at the end of the day people don't want to work for a company for free if others are being paid to work on it – "crowding out" effect – if you pay people then net reduction of work being done…."
  - Enough eyes bugs are shallow – but are the bugs interesting? – but they have to look!!
  - Institutional independence – someone can make money…do volunteers want to contribute if there is also paid labor…(crowding out) – run by a non-profit….
  - Open source from hobby to mainstream
  - Shifted narrative from "free software" to open source
- IBM – might have killing the golden goose…set up an independent organization – Linux Foundation (Red Hat, IBM etc.)
- How to structure contributions….(Netscape, red hat)
  - Rewrite code base from scratch to make contributions more simple… & community could participate..but still pay 80 engineers to work on Firefox…from the advertising base….some external contributions…make an extension framework….make a platform…e.g. pop-up blocking

- Control & decision making– e.g. "red hat legal says this won't happen"…problem

Mako Hill (cont.)

- Patent donation…software patents are considered bad in OSS
- Tension within IBM – patent licensing revenues were surpassed by patent revenues…so treat patent portfolio differently
- Foundations – gnome foundation, Mozilla foundation – hold groups of resources for volunteers….IBM partially allowed for institutional independence – important for them to have Linux be owned by IBM – people won't contribute to linux-kernel or linux/GNU if its owned by IBM but better if its a commoditized OS which gives you a collaboration
- IBM offers platforms, combination and packages….
- Canon – software that allows camera to "switch on" the ability to capture raw image data…
- OLPC – uses a Linux OS -> (US$100 laptop)…hardware is slow – no hard disc, 256MB memory, 1G flash…lighter piece of software…(faster, more efficient, pieces you can throw away…). PROBLEMS:

	GSK	Linux	D-Wave
Talent	Biologists & chemists	Software engineers	Physicists
Crowding Out	Not really	Possible that paying for work crowds out volunteerism	
Substitution			
Technical control	Exercised at Phase IIb/III	Indirect control of Linux direction – mediate via deep participation	Directed by firm over scientists
Motivation/ Thesis	Give scientists more incentives - $$ Give scientists more autonomy - < bureaucracy	Take advantage of complex motivation of "hackers" – pragmatic (p-solving), idealism, technical reputation	Scientists need resources & focus – control direction
Professional culture	Scientific community	Hacker community	Academic community

Microsoft vs. Novell

- Are Novell customers legally liable for being sued by Microsoft…because they use software that is covered by Microsoft software patents…

When & why are communities useful?

- Does open source model undermine entrepreneurial opportunities – YES! In particular spaces – things happen for free that might otherwise be costly & make money for firms.
- Same with pharmaceuticals – developing countries insisting upon breaking IP rights (but we often like this!)

BUT good lessons:

- Making money in new ways
- Providing more variety
- Making money from complementary products & services…
- Reducing the input costs for a wide range of other entrepreneurial firms