Open Source Software
Business Models Redux

Stephen R. Walli
@stephenrwalli
stephen.walli@gmail.com
There is NO Open Source Business Model

Stephen R. Walli
@stephenrwalli
stephen.walli@gmail.com
A Story
1997

~750,000 LoC

~$10,000,000
Interix
~$100,000
Interix + ACT
~$140,000
Build vs. Buy
Build vs. Buy vs. Borrow
Build vs. Buy vs. (Borrow & Share)
Open Source Software is about Engineering Economics
We’ve shared software since we’ve written software

1950
- Code sharing at Princeton IAS in late 1940s

1960
- IBM “SHARE” Conf & Library Begins 1953
- DECUS Conf & Library Begins 1962
- AT&T Shares First UNIX tapes early-70s

1970
- USENIX Begins 1975
- MIT Project Athena Begins 1983
- Emacs 1975
- 1BSD Released 1977

1980
- Free Software Foundation Launches 1985
- U.S. Congress Adds Computer Software to Copyright Law 1980

1990
- Open Source Definition 1997

2000
- OSDL Forms 2000
- OSDL Re-forms as Linux Foundation 2007

2010
- Apache Software Foundation 1999
- Apache httpd Released 1995

1956
- 1st DoJ vs IBM Consent Decree
  “Hardware Bundling is Anti-competitive” 1956

1969
- 2nd DoJ vs IBM begins
  “Software Bundling is Anti-competitive” 1969
- IBM response is to unbundle HW, SW, & services pricing

1983
- Linus Releases Linux 1991

1997
- Open Source Definition 1997

1999
- Apache Software Foundation 1999
We’ve shared software since we’ve written software
Writing good software is hard work

1950
Code sharing at Princeton IAS in late 1940s

1960
IBM “SHARE” Conf & Library Begins 1953
DECUS Conf & Library Begins 1962

1970
USENIX Begins 1975
AT&T Shares First UNIX Tapes early-70s
emacs 1975
1BSD Released 1977

1980
MIT Project Athena Begins 1983
Free Software Foundation Launches 1985
GCC 1987
Linus Releases Linux 1991
U.S. Congress Adds Computer Software to Copyright Law 1980

1990
1st DoJ vs IBM Consent Decree “Hardware Bundling is Anti-competitive” 1956
2nd DoJ vs IBM begins “Software Bundling is Anti-competitive” 1969
IBM response is to unbundle HW, SW, & services pricing

2000
Open Source Definition 1997
OSDL Re-forms as Linux Foundation 2007

2010
Free Software Foundation Launches 1985
Apache Htdp Released 1995
Apache Software Foundation 1999
OSDL Forms 2000
Linus Releases Linux 1991
GCC 1987
U.S. Congress Adds Computer Software to Copyright Law 1980
Companies shared software before we had the OSD

- Code sharing at Princeton IAS in late 1940s
- IBM “SHARE” Conf & Library Begins 1953
- DECUS Conf & Library Begins 1962
- AT&T Shares First UNIX tapes early-70s
- USENIX Begins 1975
- MIT Project Athena Begins 1983
- DEC Ultrix 1984
- SunOS 1983
- GCC 1987
- Linus Releases Linux 1991
- Red Hat 1993
- Apache httpd Released 1995
- OSF/1 1992
- Apache Software Foundation 1999
- OSDL Forms 2000
- OSDL Re-forms as Linux Foundation 2007

1950
1960
1970
1980
1990
2000
2010

- 1st DoJ vs IBM Consent Decree “Hardware Bundling is Anti-competitive” 1956
- 2nd DoJ vs IBM begins “Software Bundling is Anti-competitive” 1969
- IBM response is to unbundle HW, SW, & services pricing
- U.S. Congress Adds Computer Software to Copyright Law 1980
- Free Software Foundation Launches 1985
- Open Source Definition 1997
Companies shared software before we had the OSD (Writing good software is hard work)

- IBM “SHARE” Conf & Library Begins 1953
- DECUS Conf & Library Begins 1962
- USENIX Begins 1975
- AT&T Shares First UNIX tapes early-70s
- emacs 1975
- 1BSD Released 1977
- DEC Ultrix 1984
- SunOS 1985
- MIT Project Athena Begins 1983
- GCC 1987
- Linus Releases Linux 1991
- Red Hat 1993
- Apache httpd Released 1995
- OSF/1 1992
- Apache Software Foundation 1999
- OSDL Forms 2000
- OSDL Re-forms as Linux Foundation 2007
- U.S. Congress Adds Computer Software to Copyright Law 1980
- Open Source Definition 1997

- 1st DoJ vs IBM Consent Decree “Hardware Bundling is Anti-competitive” 1956
- 2nd DoJ vs IBM begins “Software Bundling is Anti-competitive” 1969
- IBM response is to unbundle HW, SW, & services pricing
- GCC 1987
- Linus Releases Linux 1991
- OSDL Forms 2000
- OSDL Re-forms as Linux Foundation 2007
- U.S. Congress Adds Computer Software to Copyright Law 1980
- Open Source Definition 1997
Collaboratively-Developed Liberally-Licensed Software is about Engineering Economics
Another Story
A Story of Red Hat in Three CEOs
A Story of Red Hat in Three CEOs
A Story of Red Hat in Three CEOs
A Story of Red Hat in Three CEOs
Red Hat is a software company that is focused on making its customers successful.

Red Hat is a software company that is very good at the engineering economics of open source software.
So What About the Business Model?!?!!?
The Evolution of an Open Source Project
Committer(s)
+
Code
Developers

Committers
+
Code
Building the Ecosystem

Developers

Users

Committers + Code

Books

Distributions

Support

Training

Products

Consulting

Contractors
How do you increase your user base?
How do you increase your user base?
(Because that’s where you’ll find your developers)
How do you increase your user base?
(Because that’s where you’ll find your developers)

How do you encourage developers?
(Because these are your future contributors)
How do you increase your user base?
(Because that’s where you’ll find your developers)

How do you encourage developers?
(Because these are your future contributors)

How do you make it easy to contribute?
(Because this is the growth and success of your community)
How do you increase your user base?
(How do you make it easy to install/configure/use the software?)

How do you encourage developers?
(How do you make it easy to build/test/experiment?)

How do you make it easy to contribute?
(What do you communicate to your community)
The Evolution of an Open Source Project
The Evolution of an Open Source Project
The Evolution of an Open Source Project

- Project
  - Committers
  - Contributors
  - Community
- Corporate Contributors
  - Products
  - Services
  - Books
  - Training
- Ecosystem
- Customers
Open Source Community Practices

IP Management Activities

Community Development

Software Construction Maturity
Open Source Community Patterns

Project License
Repositories Protected
Dependencies Documented
Committer Governance
Contributions Auditted
Provenance Tracking
Provenance Management
Trademark Management
Committers Indemnified

Project Exes
Project Install Automated
Project BugTracking
Complete Src published
Project Build Automated I
Project Test Automated I
Project Build Automated II
Project Test Automated II
Basic Arch Description

Encourage/Manage Corp Contributions

IP Management Activities

Mission Statement
Committer Platform
Code of Conduct
Contribution Guidelines
Governance
Events

Encourage
Community Development
Encourage Developers
Encourage Contributors

Encourage Users

Software Construction Maturity

FAQs, Howto
Forums, Email

Open Source Community Patterns

IP Management Activities

Mission Statement
Committer Platform
Code of Conduct
Contribution Guidelines
Governance
Events

Encourage
Community Development
Encourage Developers
Encourage Contributors

Encourage Users

Software Construction Maturity

FAQs, Howto
Forums, Email

Open Source Community Patterns

IP Management Activities

Mission Statement
Committer Platform
Code of Conduct
Contribution Guidelines
Governance
Events

Encourage
Community Development
Encourage Developers
Encourage Contributors

Encourage Users

Software Construction Maturity

FAQs, Howto
Forums, Email
The Evolution of an Open Source Project

Project

Committers

Contributors

Community

Corporate Contributors

Products

Ecosystem

Services

Books

Training

Customers
TRANSISION
The Evolution of a Corporate Open Source Project

Setting Customer and Partner Expectations in Community is Critical
The Reality of a Corporate Open Source Project

Setting Customer and Partner Expectations in Community is **Critical**
Customers versus Community
(Money vs. Time; Expectations are different; Conversations are different)
Customers versus Community
(Money vs. Time; Expectations are different; Conversations are different)

Partners versus Community
(Don’t mix business with community)
Customers versus Community  
(Money vs. Time; Expectations are different; Conversations are different)

Partners versus Community  
(Don’t mix business with community)

Products versus Projects  
(Success metrics are different; OKRs and KPIs are different)
The Reality of Corporate Open Source Projects

Setting Customer and Partner Expectations in Community is Critical
Collaboratively-Developed Liberally-Licensed Software is an Engineering Economic Imperative

(We’ve Shared Software Since We’ve Written Software)
Collaboratively-Developed Liberally-Licensed Software is an Engineering Economic Imperative
(We’ve Shared Software Since We’ve Written Software)

There is no Open Source Business Model
(Good Companies Solve Customer Problems)
Collaboratively-Developed Liberally-Licensed Software is an Engineering Economic Imperative
(We’ve Shared Software Since We’ve Written Software)

There is no Open Source Business Model
(Good Companies Solve Customer Problems)

Successful Open Source Projects Follow Well Known Patterns
(No Scale without Discipline)
Collaboratively-Developed Liberally-Licensed Software is an Engineering Economic Imperative  
(We’ve Shared Software Since We’ve Written Software)

There is no Open Source Business Model  
(Good Companies Solve Customer Problems)

Successful Open Source Projects Follow Well Known Patterns  
(No Scale without Discipline)

Don’t Confuse Customers, Partners, Community, Products & Projects  
(Corporate Open Source Projects Require Different Thinking)
How do you make money when you give away the product for free?
How do you make money when you give away the product for free?
What problem are we solving for our customers?
What problem are we solving for our customers?
How do we make our customers successful?
What problem are we solving for our customers?
How do we make our customers successful?
What can we do to make our community healthier?
What problem are we solving for our customers?
How do we make our customers successful?
What can we do to make our community healthier?
How do we make our community successful?
No Scale without Discipline
(This is an Attribute of Software)

Orders of Magnitude
(1000:100:10:1)

Freeloaders are important!
(It means you’re doing it right)