OUTERcurve
FOUNDATION

Open Source, Business, and Mobile

Stephen R. Walli
Outercurve Foundation Technical Director
Twitter: @stephenrwalli

30 March 2011

Free and Open Source Software

Source code for the software is always available, and the user of the software is liberally licensed (always) to change and redistribute the software without fee, penalty or asking permission

OUTERCURVE.ORG

We have shared software since we have written it

OUTERCURVE.ORG

DECUS (1970s-80s)

SHARE (1950s-80s)

Institute for Advanced Learning (1940s)

Two Ratios

~20 LoC / day

1 bug : 1000 LoC

OUTERCURVE.ORG

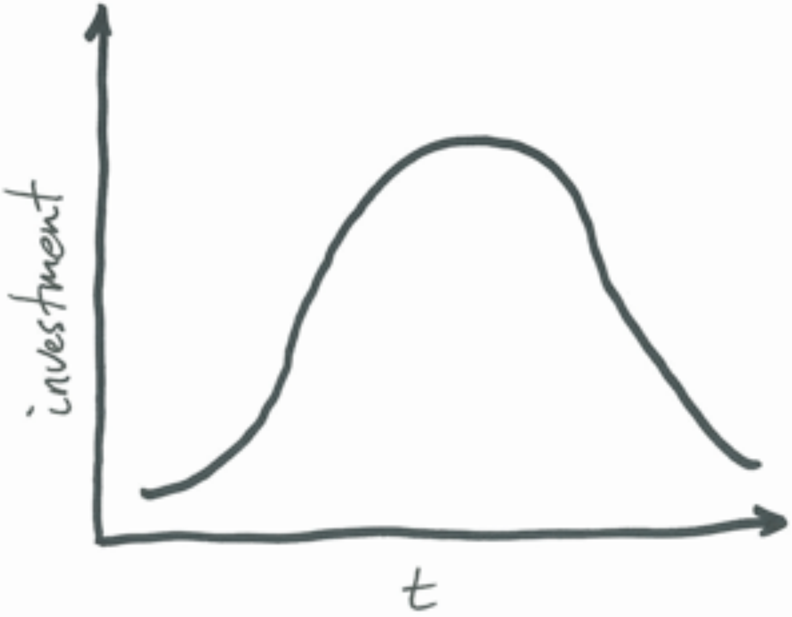
Plagiarism in software is good and indeed necessary

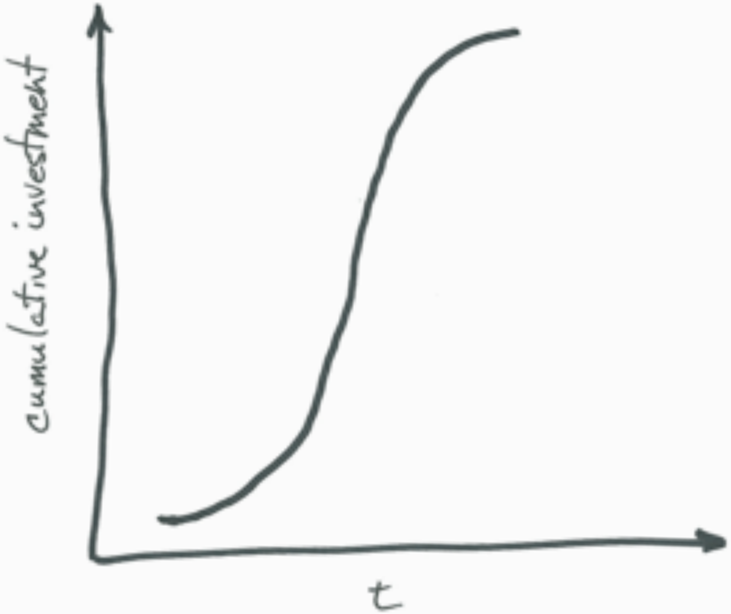
Two More Ratios

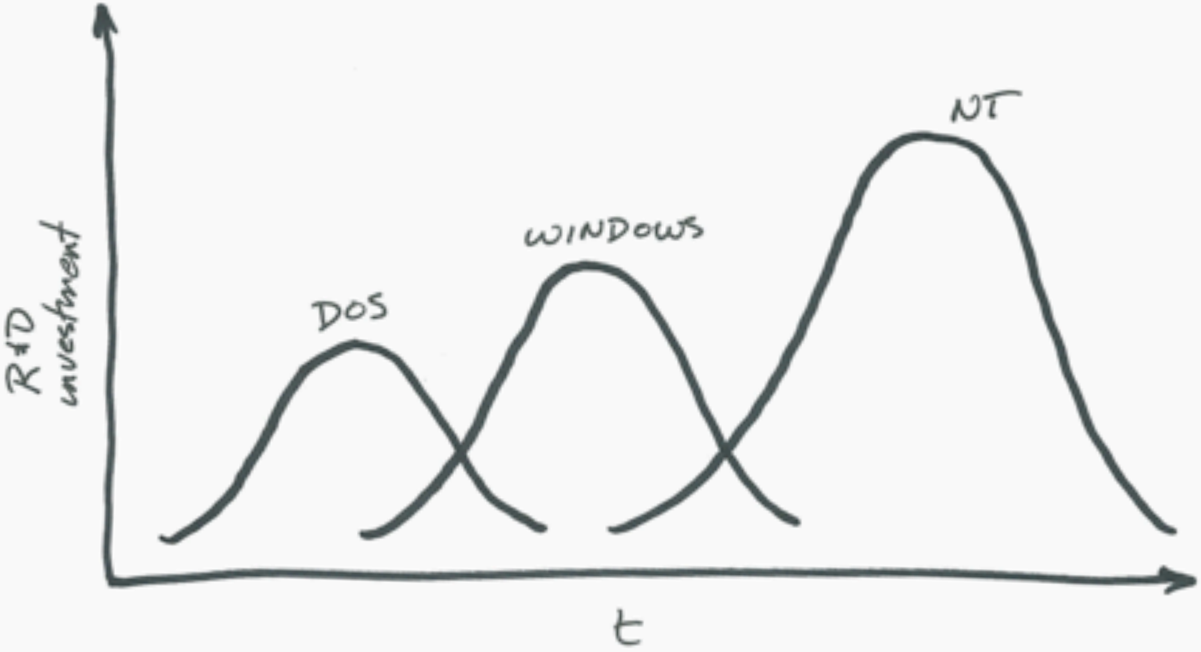
75% of an applications cost is “maintenance”

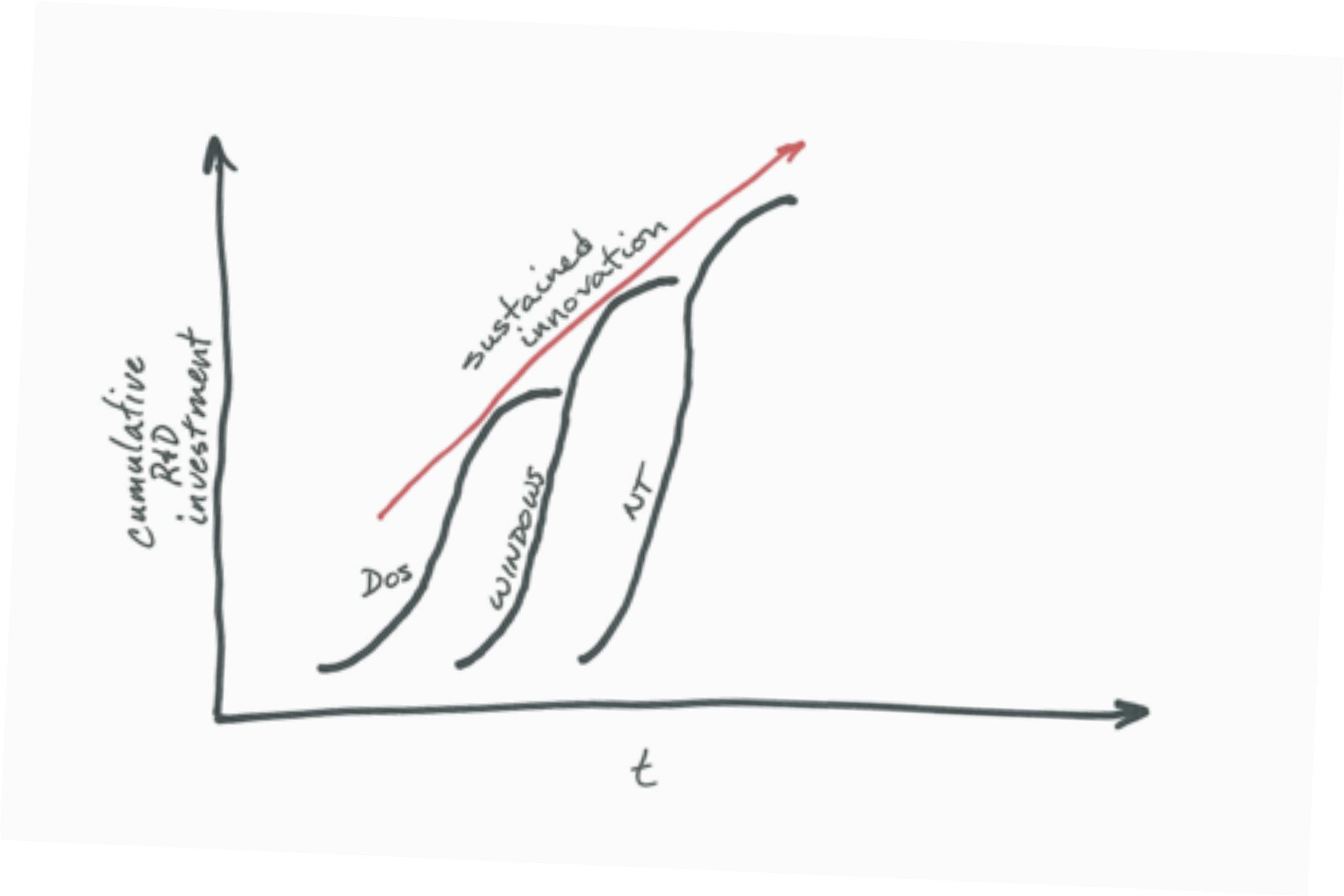
60% of applications never get deployed

Proprietary Software WAS Good for You


investment
t


cumulative investment
t


R&D investment
DOS
WINDOWS
NT
t


Cumulative R+D investment
sustained innovation
DOS
WINDOWS
NT
t

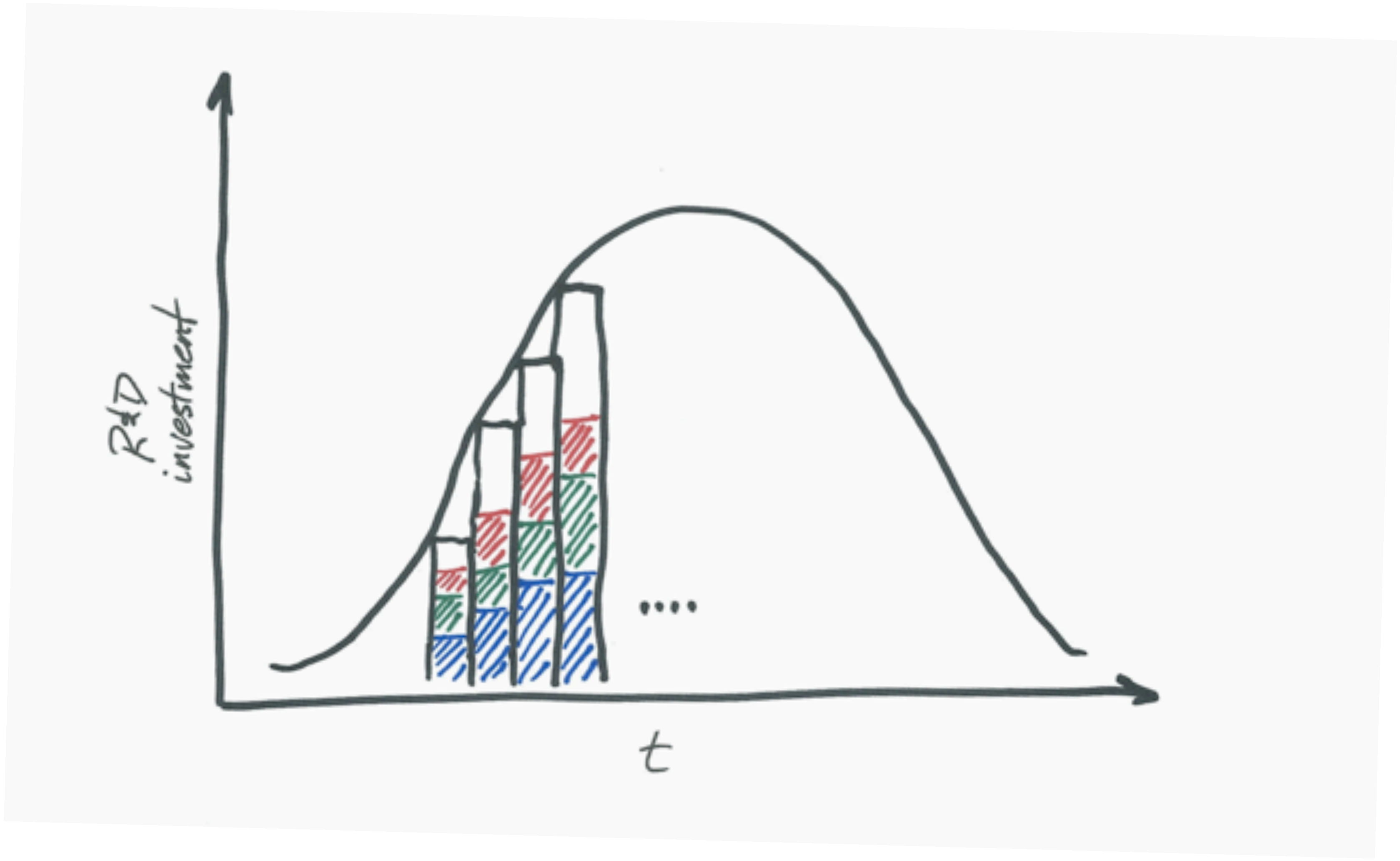
An then the Internet Happened …

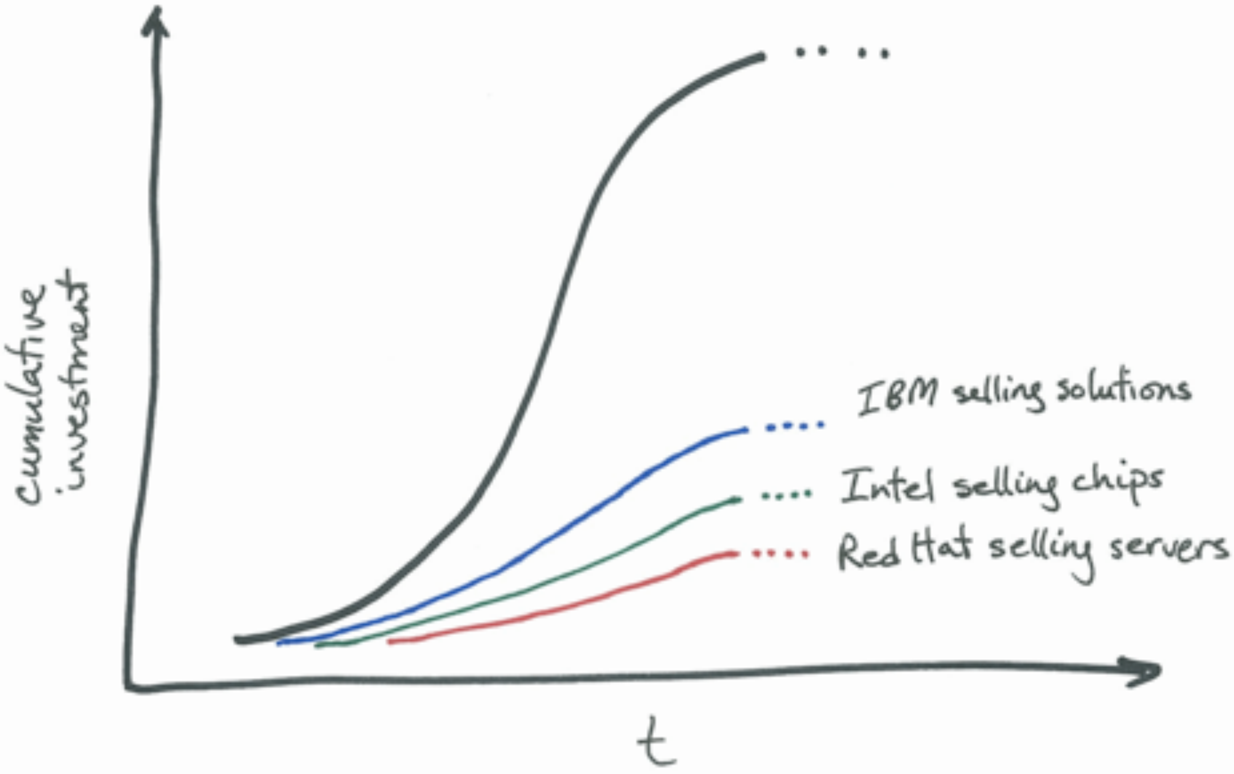
Tape-sized packets and conference latency
(or 300 baud modems)

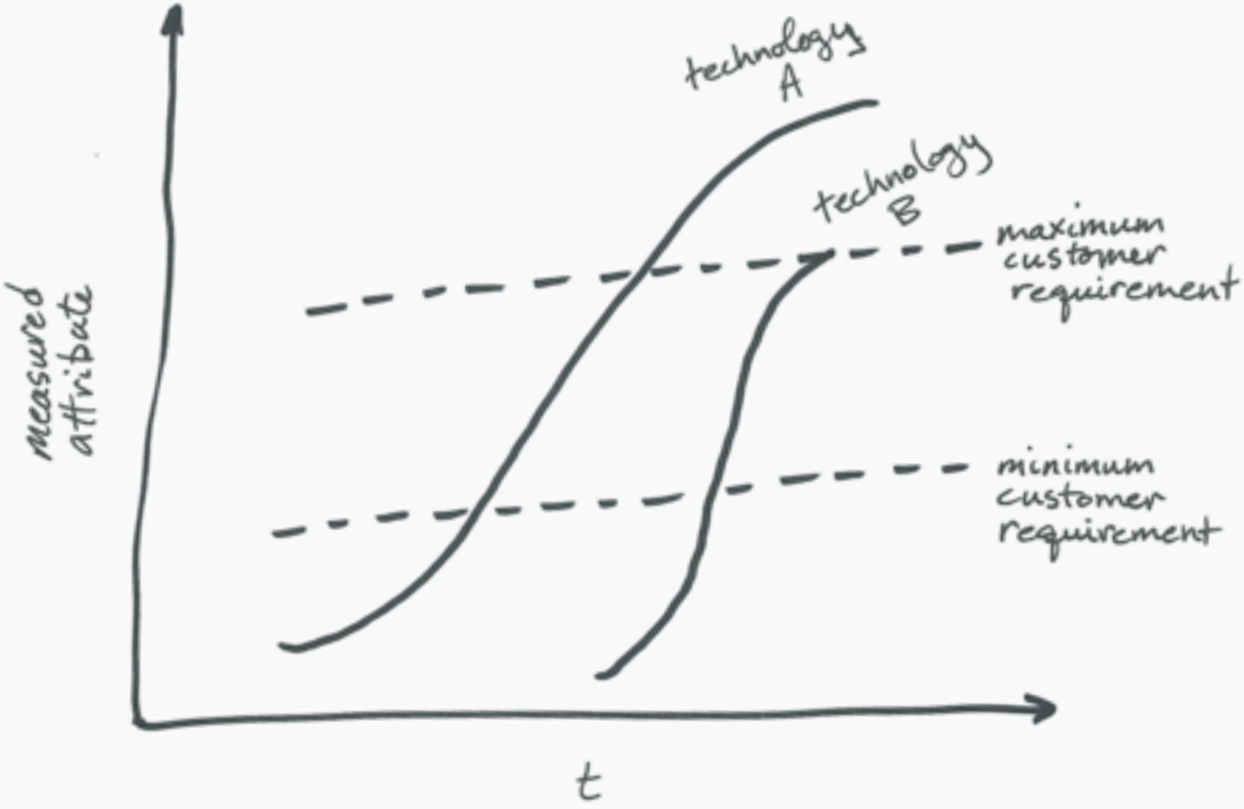
versus

The Internet and the WWW

OUTERCURVE.ORG


R&D investment
....
t

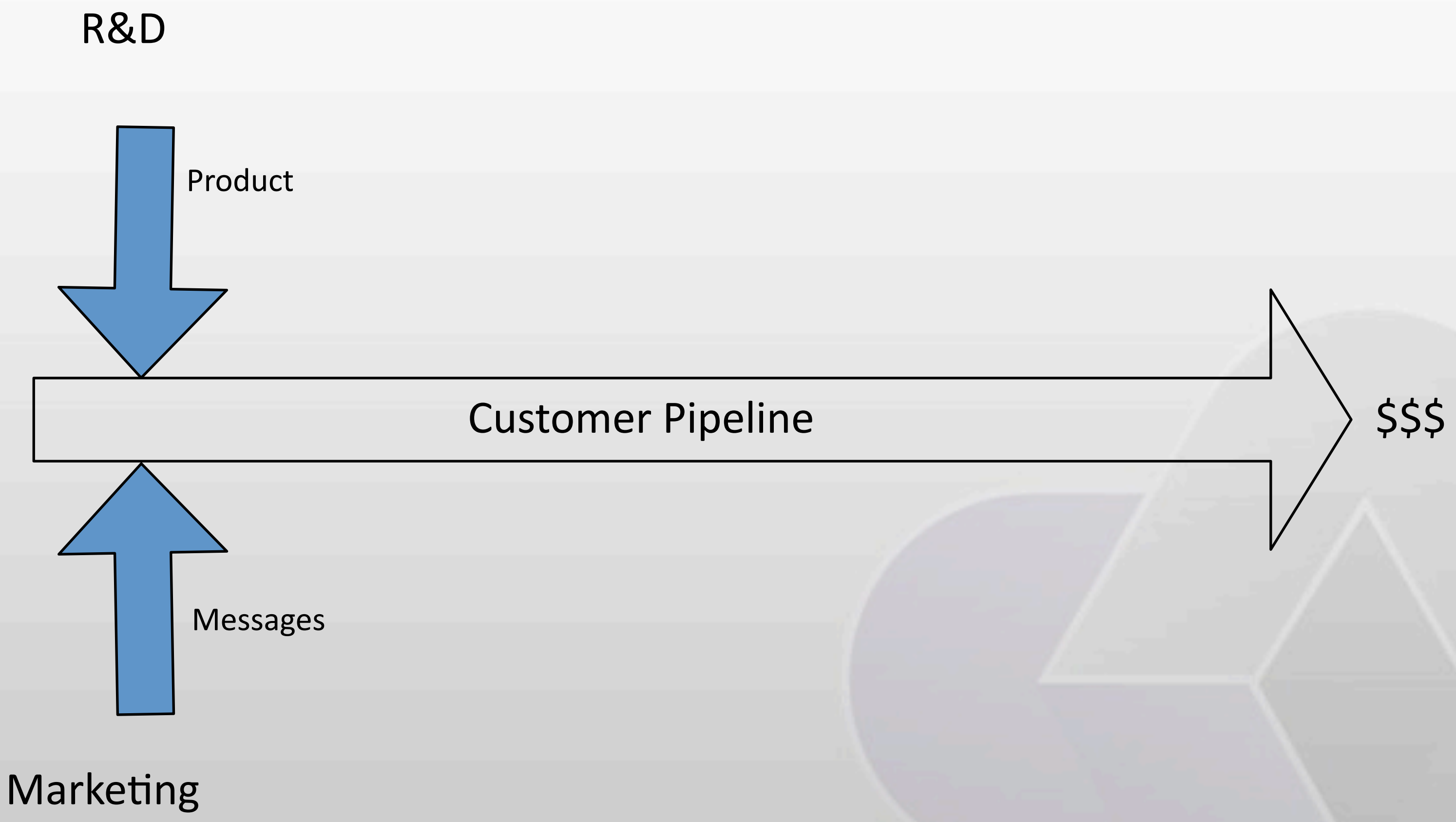

cumulative investment
IBM selling solutions
Intel selling chips
Red Hat selling servers
t


measured attribute
technology A
technology B
maximum customer requirement
minimum customer requirement
t

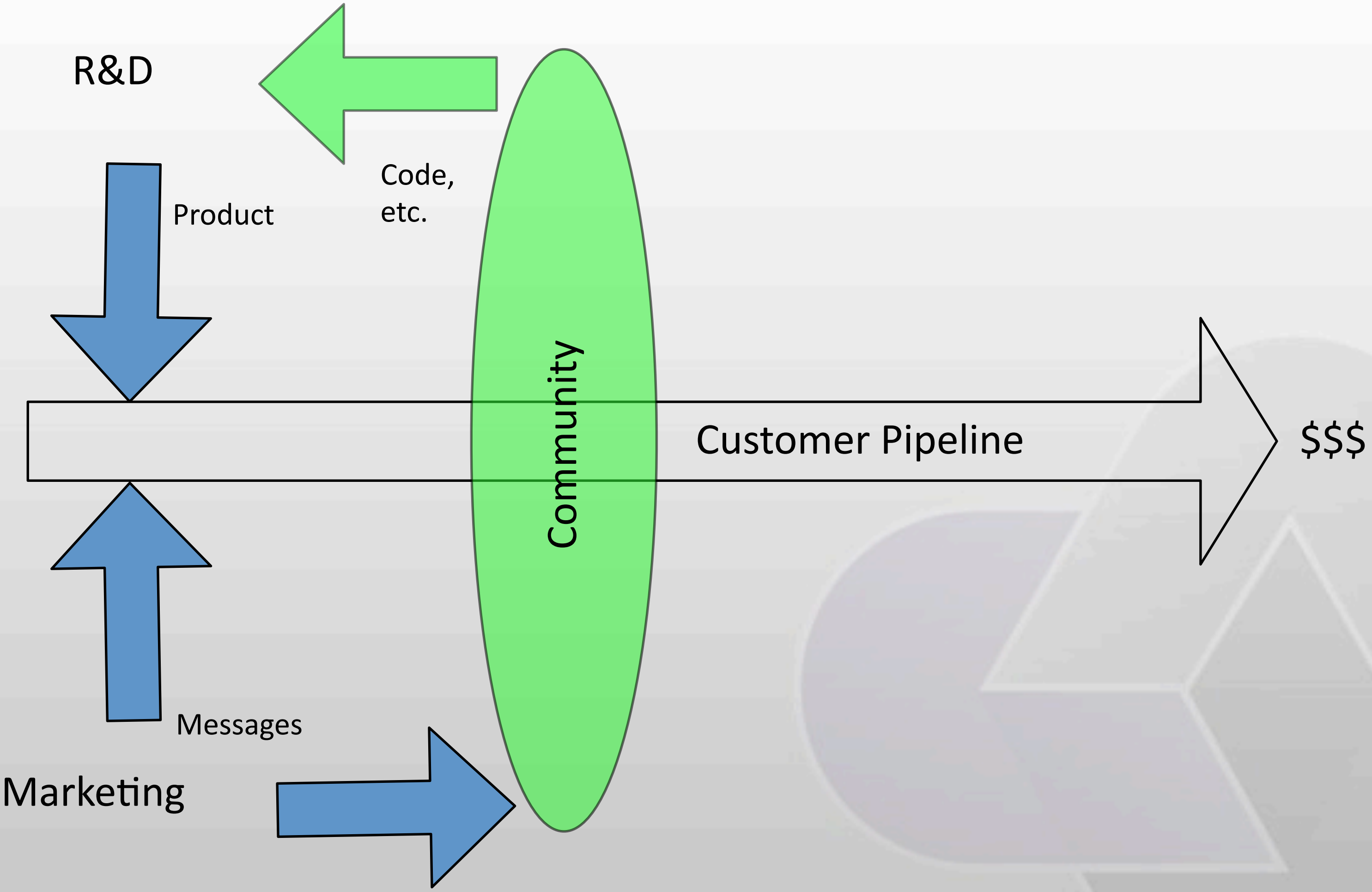
The Campfire Rule

We've understood communities since you had a campfire and I wanted to sit beside it

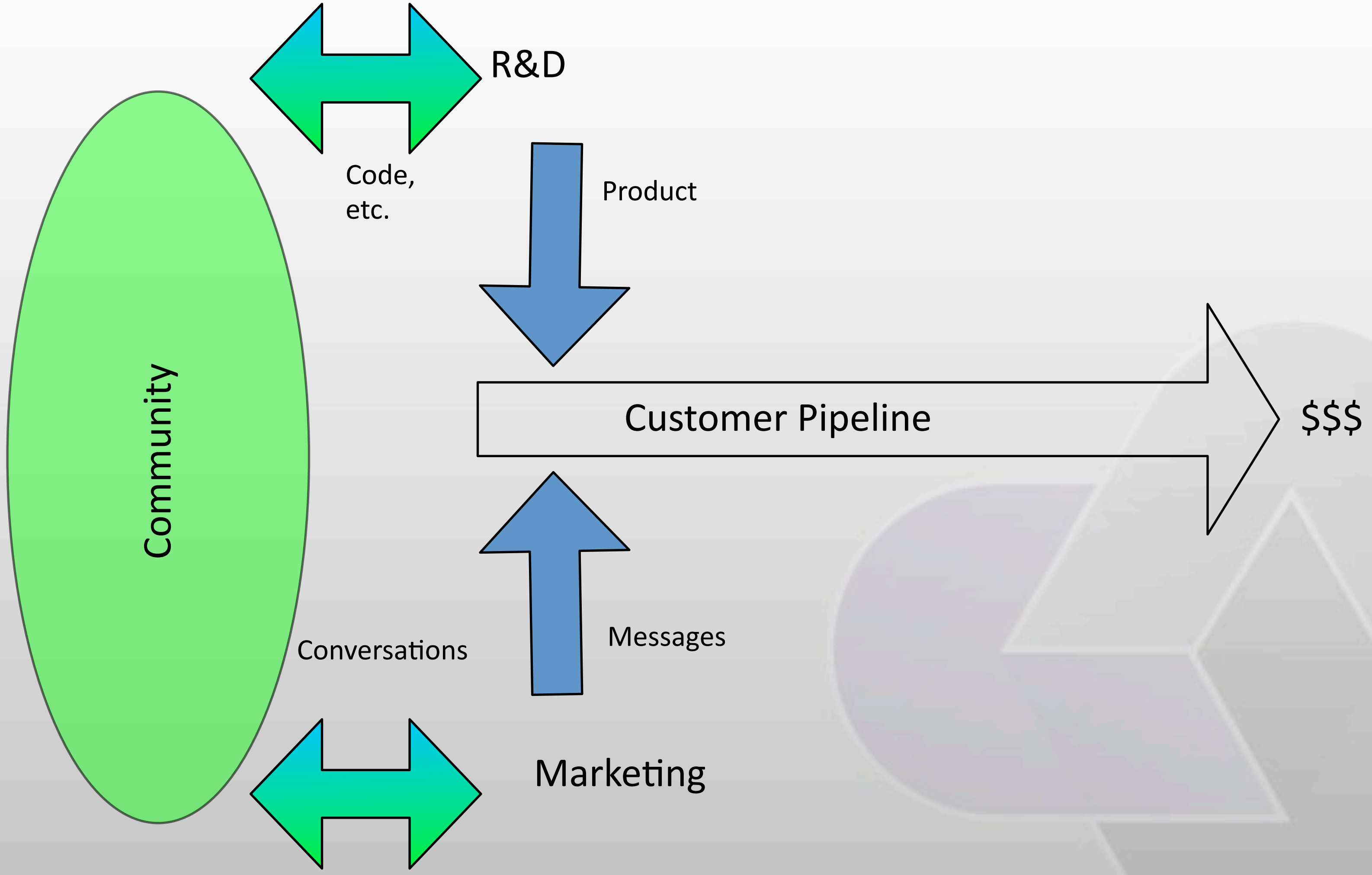
Traditional Software Business



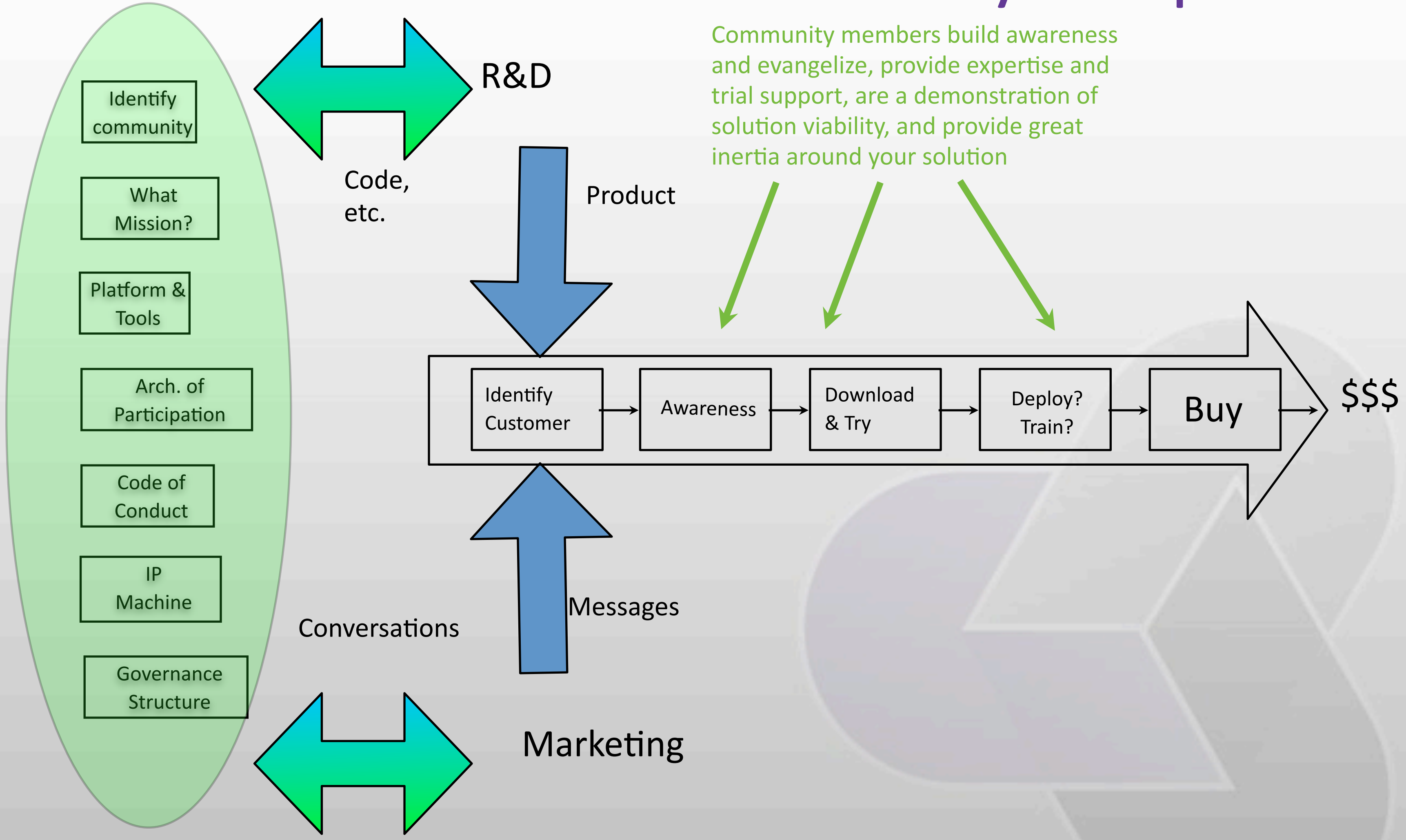
Misconceptions about Community



A Better Model for Community & Pipeline



A Better Model for Community & Pipeline



Identify community

What Mission?

Platform & Tools

Arch. of Participation

Code of Conduct

IP Machine

Governance Structure

R&D

Code, etc.

Product

Community members build awareness and evangelize, provide expertise and trial support, are a demonstration of solution viability, and provide great inertia around your solution

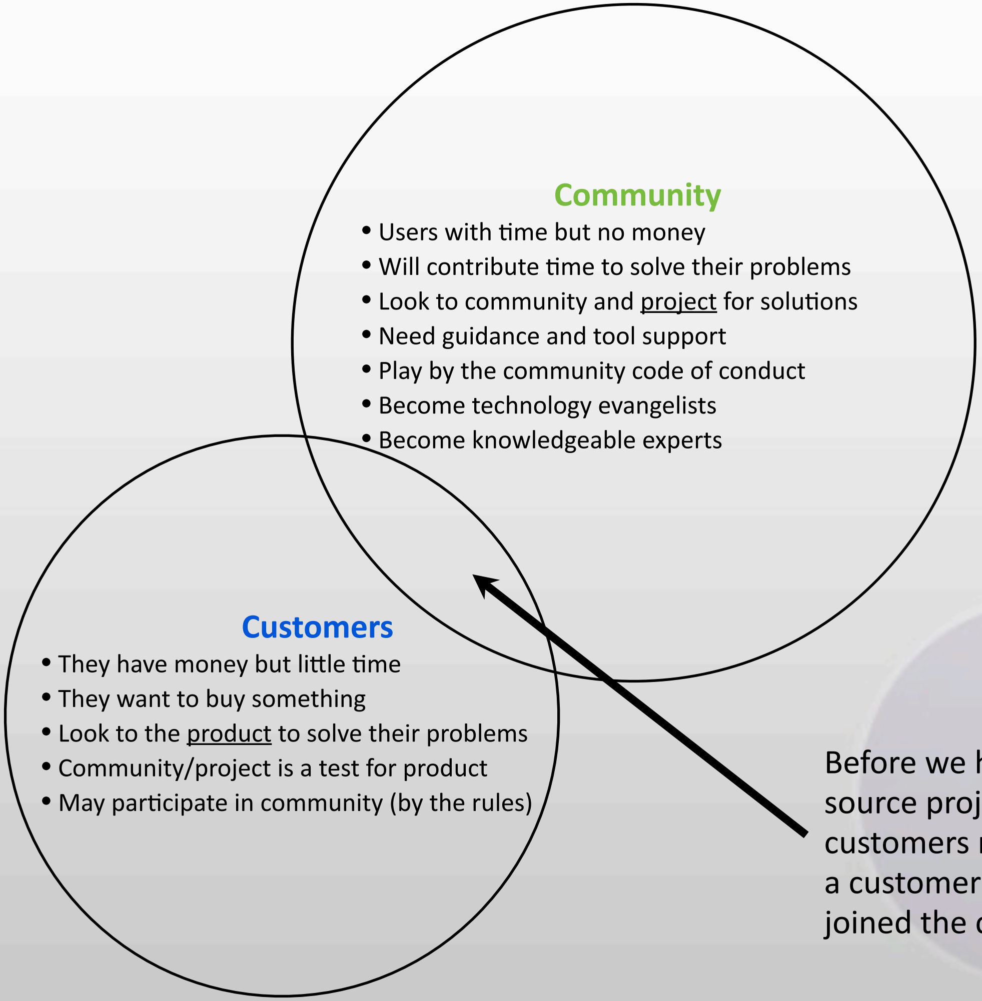
Identify Customer → Awareness → Download & Try → Deploy? Train? → Buy → $$$

Messages

Conversations

Marketing

Understanding Community & Customers



Community

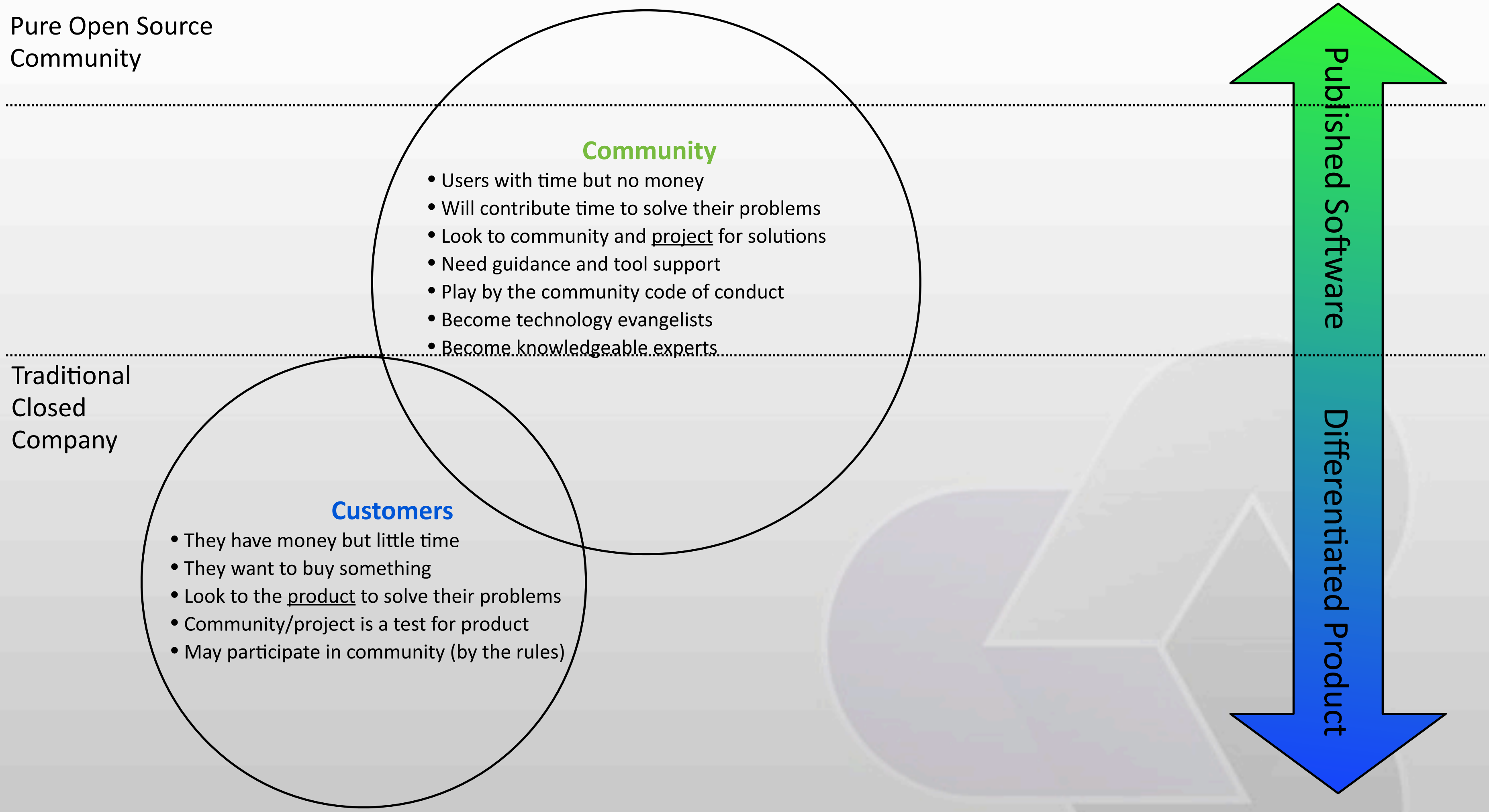
- Users with time but no money
- Will contribute time to solve their problems
- Look to community and project for solutions
- Need guidance and tool support
- Play by the community code of conduct
- Become technology evangelists
- Become knowledgeable experts

Customers

- They have money but little time
- They want to buy something
- Look to the product to solve their problems
- Community/project is a test for product
- May participate in community (by the rules)

Before we had online communities around open source projects, tech communities overlapped customers much more because one needed to be a customer before one had the interest and joined the community

Understanding Community and Open Source



Pure Open Source Community

Traditional Closed Company

Community

- Users with time but no money
- Will contribute time to solve their problems
- Look to community and project for solutions
- Need guidance and tool support
- Play by the community code of conduct
- Become technology evangelists
- Become knowledgeable experts

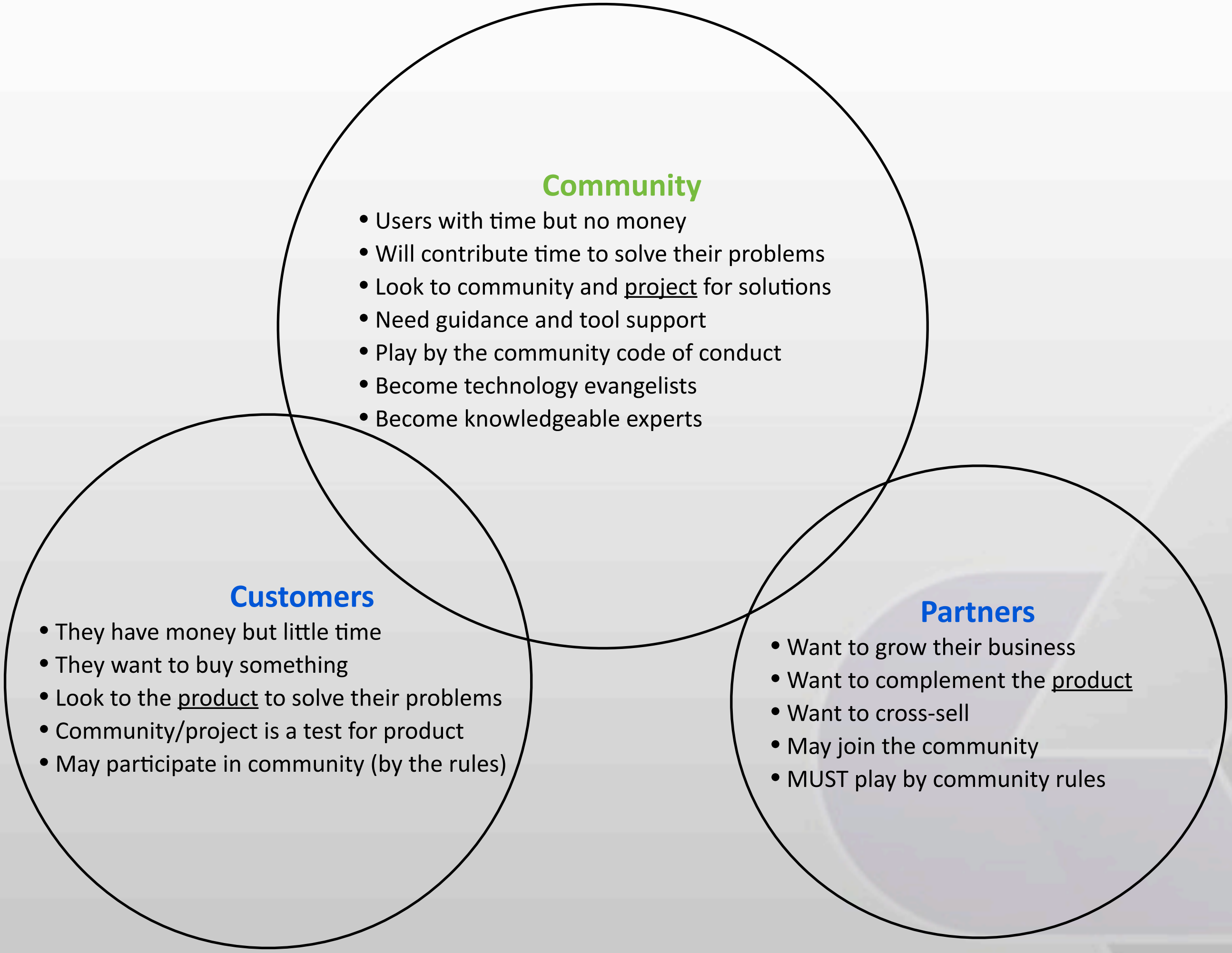
Customers

- They have money but little time
- They want to buy something
- Look to the product to solve their problems
- Community/project is a test for product
- May participate in community (by the rules)

Published Software

Differentiated Product

Understanding Community & Partner Programs



Community

- Users with time but no money
- Will contribute time to solve their problems
- Look to community and project for solutions
- Need guidance and tool support
- Play by the community code of conduct
- Become technology evangelists
- Become knowledgeable experts

Customers

- They have money but little time
- They want to buy something
- Look to the product to solve their problems
- Community/project is a test for product
- May participate in community (by the rules)

Partners

- Want to grow their business
- Want to complement the product
- Want to cross-sell
- May join the community
- MUST play by community rules

Projects to Products

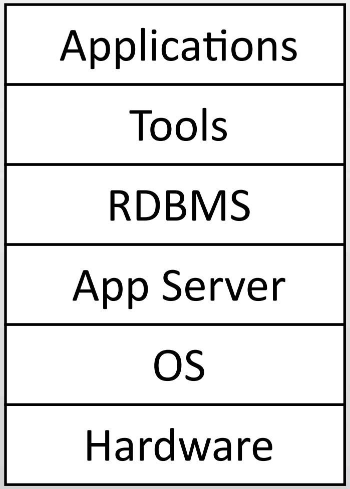
- A product is packaged, installable, tested, documented, supported, and maintained for customers; money changes hands and there is a customer expectation

- Companies build products as part of their value proposition to their customer; another way to say this is customers buy products (solutions), not software

Crossing the Chasm

- A classic book by Geoff Moore written in 1991
- The adoption curve for a product starts with early adopters, then the early majority, the late majority, and the "luddites" — there is a gap between early adopters and the early majority (the "chasm") and it's based on risk
- He defines the idea of the whole solution for the customer (i.e. your "core" and all its complements)
- Core here is the core value proposition – it is by definition customer facing
- There are a collection of strategies that companies develop to produce the "whole solution"

It’s not a stack – Open your thinking



Applications

Tools

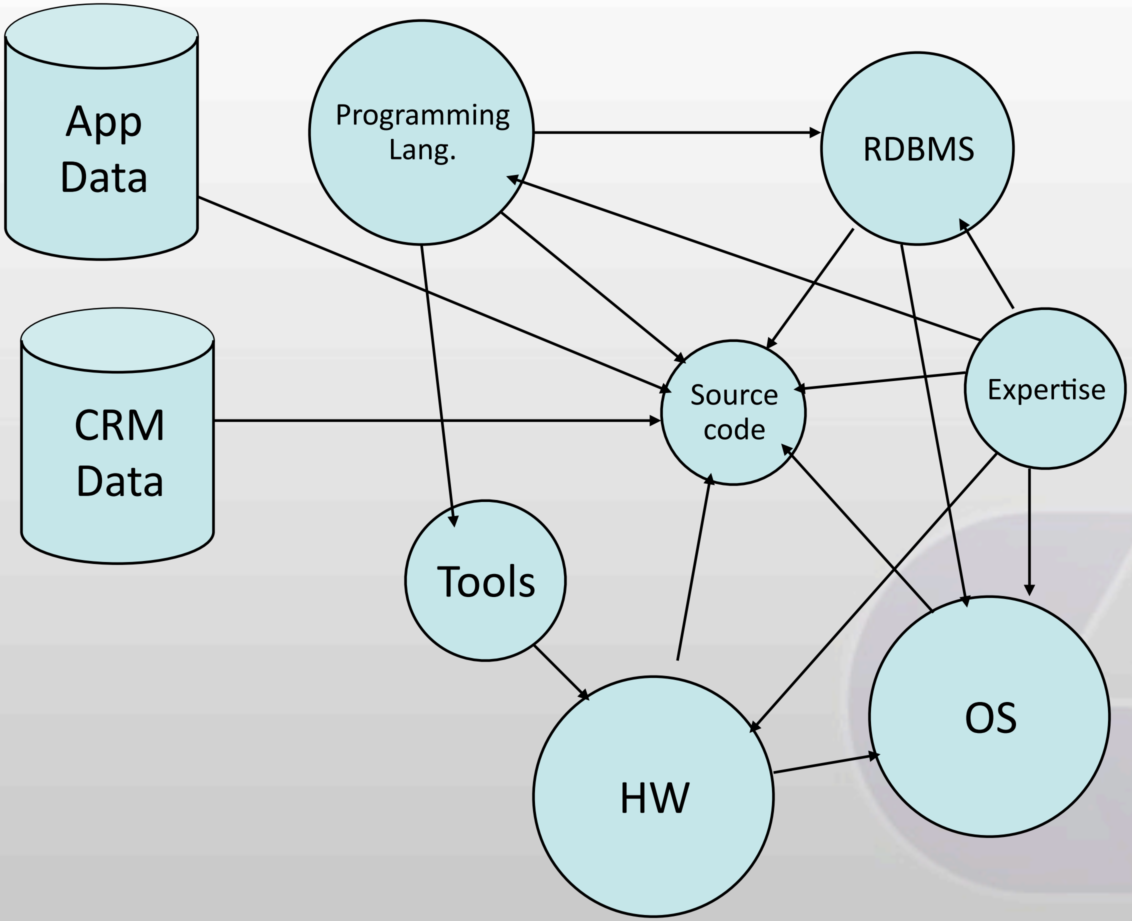
RDBMS

App Server

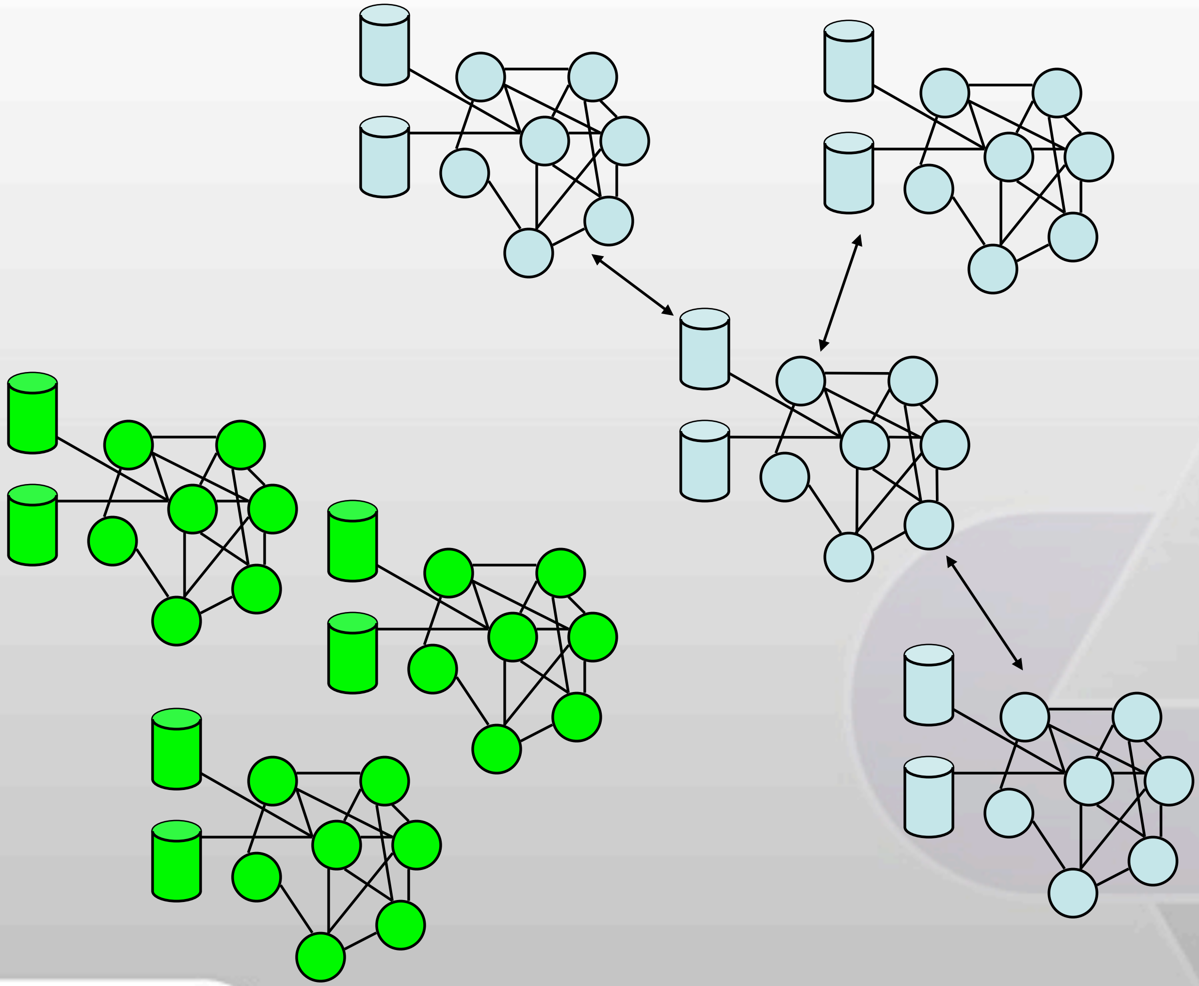
OS

Hardware

The application is actually a network …

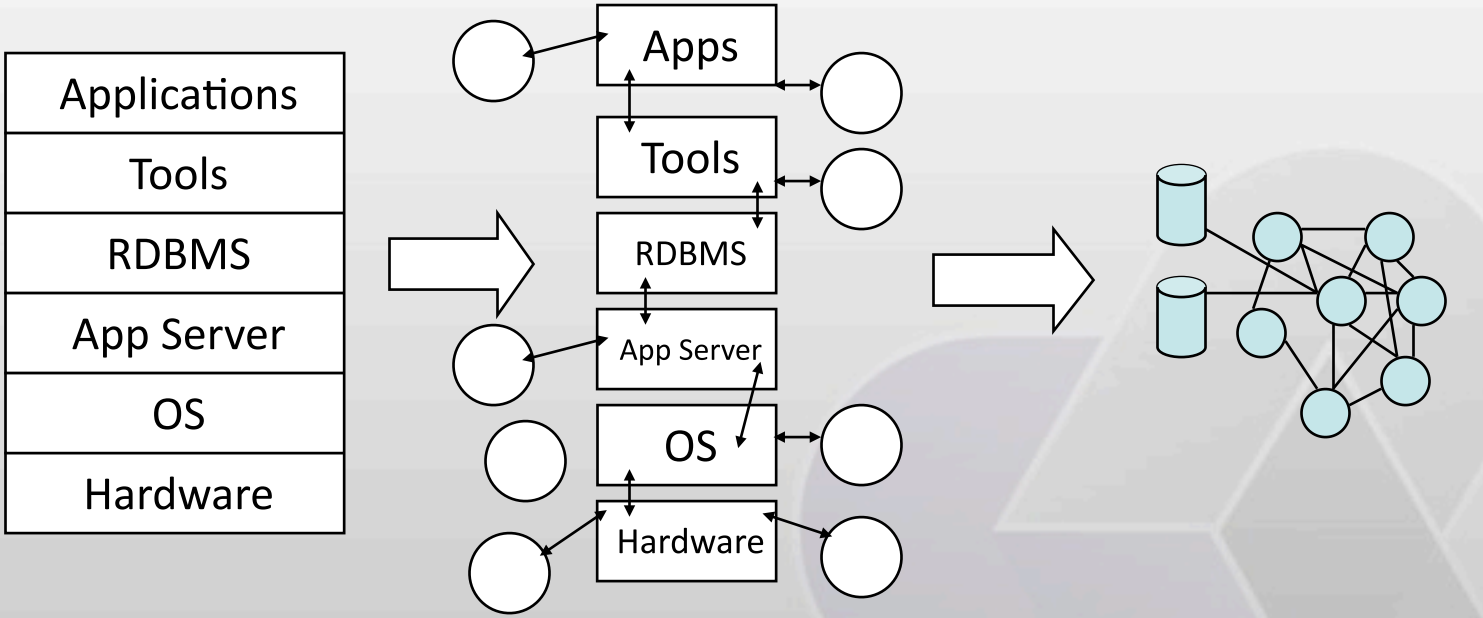


… and the network isn't "simple"



OUTERCURVE.ORG

It’s not a stack – it’s a network.



Applications

Tools

RDBMS

App Server

OS

Hardware

Apps

Tools

RDBMS

App Server

OS

Hardware

i.e. the “stack” is a view through the network.

“Whole Solution” Business Tools (for Vendors)

- Traditional buy-versus-build strategies through the vendor’s own brand, regardless of whether the complement products are offered as add-ons or bundled directly with the core revenue stream for “free”
- Develop a rich ecosystem of add-ons by encouraging developer and partner networks to provide a bigger whole solution
- Publish proprietary specifications enabling more partners to develop stable businesses in the complement spaces (N.B. this is NOT a standard – as much as the vendor will claim it to be)
- Developer tools that help add complements to the ecosystem
- Certification programs around the core technology creating service professionals to help customers complete and support their solution
- Certification programs around the core technology to identify products that “work together” with the core
- Develop a consulting services arm for part of the solution
- Develop training programs and train-the-trainer certifications

F/OSS Business Tools

- To buy-vs-build you add “borrow” and “share”
- Companies that participate in communities
  - Can rapidly develop complements to core offerings in their solution network (without necessarily building complete products)
    - SAP and MAXDB from MySQL
  - Can amortize dev/support/maint costs of software components across customers/partners/competitors
    - IBM and Apache and Websphere
    - What Oracle *should* do with JBoss
    - The vendors in the Linux Foundation today are no different than the vendors in the OSF 20 years ago sharing the development costs of OSF/Motif and OSF/1 as royalty free base technology
  - Get to interact directly with like-minded customer prospects in community, influencing customer/partner developers
- Companies that participate deeply in communities better influence those communities (e.g. participate, hire, or acquire)

OUTERCURVE.ORG

F/OSS Business Tools - 2

- Companies can use F/OSS projects to reduce the cost of sales by allowing users to try easily and pre-qualify themselves as customers
- F/OSS projects are an interesting publication strategy against competitors from an IP strategy perspective
- New companies with lower margin business models (compared to the incumbent) can use F/OSS components to rapidly develop products that either serve new markets or serve the bottom end of an over-served market and then evolve over to or up into an incumbent’s market space

OUTERCURVE.ORG

Dealing with Darwin

- Geoff Moore's “new” classic (2006)
- Core is defined as “any process that contributes directly to the sustainable differentiation leading to competitive advantage in target markets”
- Context is defined as “all other processes required to fulfill commitments to one or more stakeholders”
- WARNING: he's now defining core as core competency – this is DIFFERENT from core value proposition
- This core is not customer facing – it is about the competitive advantage that enables the core value proposition

Value Proposition versus Competency

- Microsoft
  - their core value proposition is an incredible business desktop appliance
  - their core competency is delivering packaged software that works out of the box across the entire hardware compatibility list crossed with the application compatibility list
- Google
  - their core value proposition is a great advertising engine
  - their core competency is the ultimate search machine

The "Rules"

- Understand the difference between the value proposition to your customer (i.e. the business model) and what competencies get you there
- Understand your value proposition is WHY customers give you money – it is hopefully not different to your "messaging"
- NEVER share/publish your core competency (e.g. Windows versus Google or a crypto company)
- People will always pay for value

The Business of Community Building

OUTERCURVE.ORG

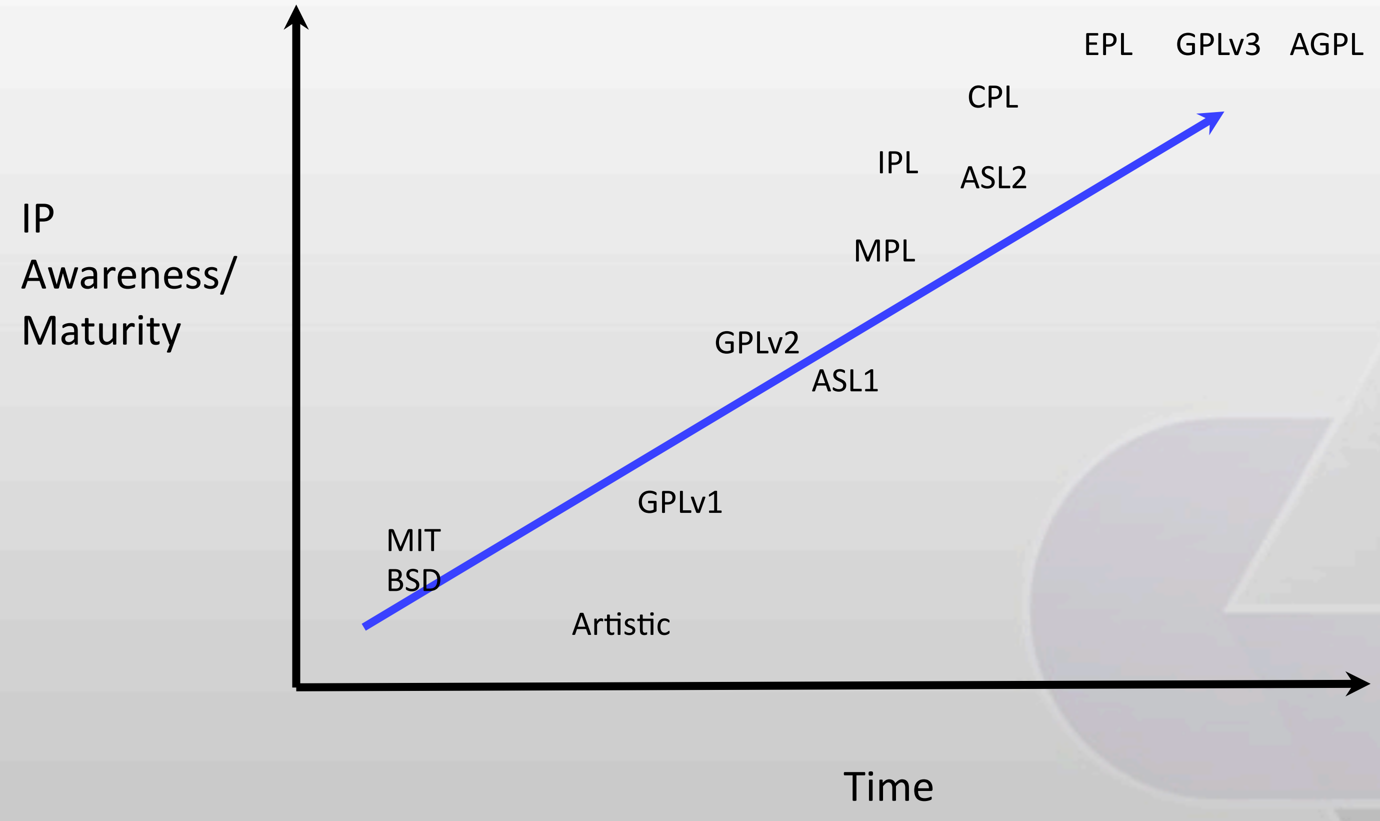
A Quick Summary of the Rules

- Create an Architecture of Participation
  - Start the conversation with code and great technology
  - Have frequent releases
- Need to make it easy to join the conversation
  - Give people things to do
  - Build tutorials, documentation, books
- Find and support your tribal leader
- Committers need to be strong communicators with good conflict resolution skills
- Be as transparent as possible
  - No internal mailing lists
  - Publish the bug database
  - Push everything to the edge

A Quick Summary of the Pitfalls

- Not all developers are good communicators
- Marketers still want to be in control
- Too easy to fall back into the old way of doing things
- If you publish it the world will NOT beat a path to your door (Mozilla)
- Holding back technology (Hula)
- Holding back information is a breach of trust

The Maturation of OSS Licensing



OUTERCURVE.ORG

The Rise of the Foundation

- FSF and the evolution of software freedom
- Apache project to the ASF
- Mozilla to Mozilla Foundation & Mozilla Inc.
- Eclipse project to the Eclipse Foundation
- OSDL + Free Standards = Linux Foundation
- Gnome project to the Gnome Foundation

Open Source Software Foundations

All provide:

- IP Management Machine
- Project Management Discipline (macro/micro)
- Marketing, Outreach, PR

Some provide:

- Forge
- Distribution

Symbian Foundation Failure

- No Neutrality: First amongst “equals” and code flow
- No Architecture of Participation for technology roll-out
- No responsibility to community – committers have names
- Role Confusion and Staffing Problems
- 100 Employees versus 25 versus 2+

Questions

- Stephen R. Walli, Technical Director, swalli@outercurve.org
  - http://stephesblog.blogs.com
  - http://www.networkworld.com/community/walli
  - Twitter @stephenrwalli

vs.

TM
open source

Software Economics and Open Source

OUTERCURVE.ORG

Communities (and Customers)

OUTERCURVE.ORG

Projects and Products

The Mechanics of Building a Strong Foundation

- Roles
- Set and publish expectations
- Build the IP management machine [**consider a foundation**]
- Develop the rules for how plans will be made in the project
- Develop the infrastructure or choose the forge

Build the IP Machine

- Everyone needs to understand the rules with respect to contributions and IP cleanliness
- Put a process in place and publish it
- Ensure all IP contributions have the proper “paper work” with respect to the contributor and their employer
- Ensure committers understand their responsibilities and authority with respect to IP

Fear, Uncertainty, and Doubt

- Why can't you push everything to the edge?
- What needs to be "secret"? Why?
- What mistake could be made that is unrecoverable? (I'm betting there isn't one)
- Ultimately you still have code (committer) control while you learn who else has merit
- Who is your competition?

Building Foundations

OUTERCURVE.ORG

Types of OSS Non-Profits

- Examples
- Positive Attributes
- Negative Attributes

Sponsor Driven "DIY"

- Fedora, OpenSuSe
- Sponsor has highest level of control
- Focused projects
- Well understood brand
- Sponsor Primary source of funds
- Narrow technology focus
- Smaller potential community
- Responsible for complete process

Mix of Sponsors and Community

- Linux Foundation
- Eclipse Foundation
- Python Foundation
- Inclusionary
- Revenue mix not tied to one member or sponsor
- Broader technology focus
- More complex operationally
- May dictate license and/or development methodology
- Expensive

Community Driven

- Apache
- Gnome
- Inclusive
- Low operating costs
- Strong volunteer ethic
- Less disciplined
- Dependent on volunteers for admin
- Less structure
-

The Outercurve Foundation

What are corporations looking for?

- Liability Firewall
- Demonstration of Neutrality
- Useable IP
- Foundation in a Box

What issues do we address?

- Most Corporations do not want the risk/ liability of creating an OS project
- Move the risk to a non profit
- Shift the perception of who benefits
- Commercial organizations that want to leverage shared development need to know the provenance of the IP
- Foundations require significant startup expense, staffing, legal, and governance.

Benefits

- Outercurve is responsible for the IP and the IP management process
- Everyone understands and benefits from the shared development

- Outercurve manages assignment and licensing process
- Outercurve ensures development process maps IP process to track provenance.

- Outercurve provides staffing admin, and operational expertise to deliver the functions of a .org

OUTERCURVE.ORG

The Outercurve Foundation

While many Foundations provide a liability firewall and a demonstration of neutrality

- ➢ They are tied to a license, a particular technology base, or development process.

The **Outercurve Foundation** provides the benefits of neutrality, a liability firewall, and well packaged IP, at a low cost

- ➢ Remaining agnostic on license, technology platform, and development process.

Our Mission

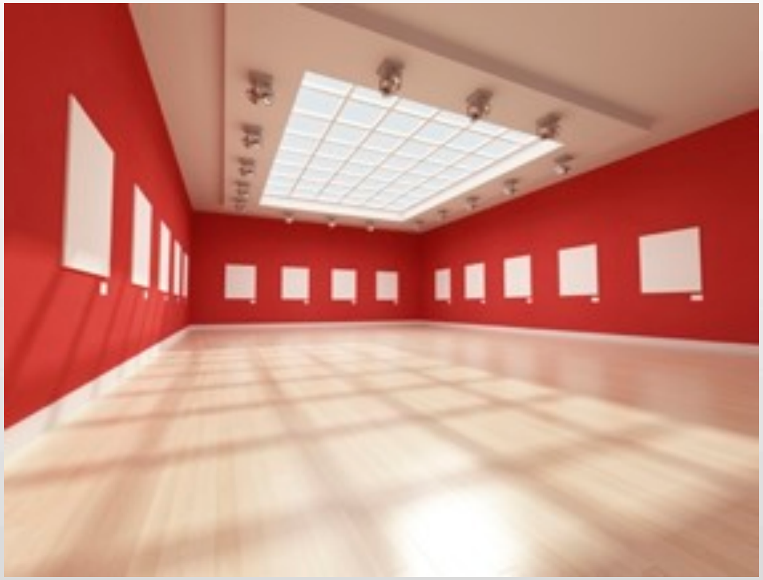
Enable the exchange of code and understanding among software companies and open source communities.

Multi-source Vendors

OSS Communities

Corporate IT

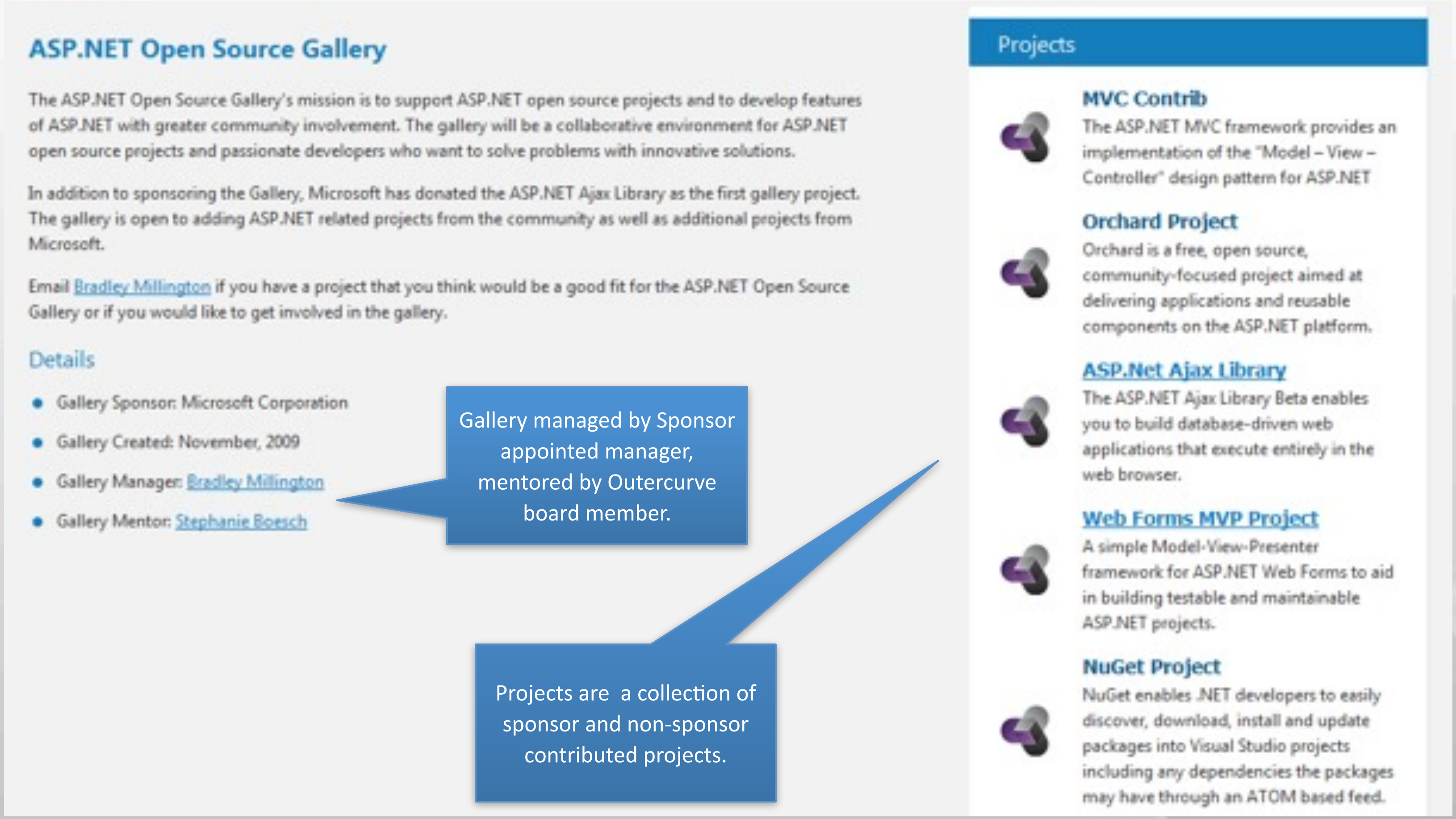
The Museum As A Metaphor



- **Sponsors** create “Galleries” based on technology or industry themes
- **Gallery Managers** and the Foundation encourage project assignments into Galleries
- **Individual Projects** are complementary with the theme of the Gallery

OUTERcurve.ORG

Gallery Example
Technology Themed: ASP.NET



ASP.NET Open Source Gallery

The ASP.NET Open Source Gallery's mission is to support ASP.NET open source projects and to develop features of ASP.NET with greater community involvement. The gallery will be a collaborative environment for ASP.NET open source projects and passionate developers who want to solve problems with innovative solutions.

In addition to sponsoring the Gallery, Microsoft has donated the ASP.NET Ajax Library as the first gallery project. The gallery is open to adding ASP.NET related projects from the community as well as additional projects from Microsoft.

Email Bradley Millington if you have a project that you think would be a good fit for the ASP.NET Open Source Gallery or if you would like to get involved in the gallery.

Details

- Gallery Sponsor: Microsoft Corporation
- Gallery Created: November, 2009
- Gallery Manager: Bradley Millington
- Gallery Mentor: Stephanie Boesch

Projects

MVC Contrib
The ASP.NET MVC framework provides an implementation of the "Model – View – Controller" design pattern for ASP.NET

Orchard Project
Orchard is a free, open source, community-focused project aimed at delivering applications and reusable components on the ASP.NET platform.

ASP.Net Ajax Library
The ASP.NET Ajax Library Beta enables you to build database-driven web applications that execute entirely in the web browser.

Web Forms MVP Project
A simple Model-View-Presenter framework for ASP.NET Web Forms to aid in building testable and maintainable ASP.NET projects.

NuGet Project
NuGet enables .NET developers to easily discover, download, install and update packages into Visual Studio projects including any dependencies the packages may have through an ATOM based feed.

Gallery managed by Sponsor appointed manager, mentored by Outercurve board member.

Projects are a collection of sponsor and non-sponsor contributed projects.

OUTERCURVE.ORG

Foundation Operations and Project Acceptance

By-laws

- The ground rules established by the Board of Directors for the governance of the foundation
  - Seats available to Corporate Sponsors
- Staff manages day to day operations and administrative functions

Project acceptance and operations guidelines

- Serves as a policy for how projects are accepted into the Foundation, and how they are organized and managed once accepted
- Gallery Managers oversee projects within their Gallery
  - Gallery Manager appointed by Gallery

Assignment agreement

- Serves as a template for how companies or individuals assign copyright or license patents to the Foundation

Contribution agreement

- Serves as a template for how companies or individuals may license copyrights or patents to the Foundation

Outercurve Provides Staffing and Services Throughout the Development Lifecycle

Concept
- Counsel and content on best practices for licensing, code assignment, and contribution agreements

Launch
- Launch program tailored to the project audience(s), including integrated PR and Social Media campaigns

Build Community
- Establish project management process

Sustain Support
- Gallery and project sponsors provide 3 year commitment

Adoption
- Provide tools and templates to guide project leads to successful adoption of the project

Why Sponsor an Outercurve Gallery?

A strategic decision	• Accelerate open source participation within your organization • Encourage contributions to key projects by distancing yourself from the IP • Ensure ongoing contribution, resulting in greater adoption • Extend your platform with open source projects • Protect your IP
Assert your thought leadership	• Be a sponsor of important work • Partner with industry leaders as they adopt open source
Reduce expense	• Expend fewer legal resources once project is assigned to the Foundation • Cost effective alternative to starting your own non

Galleries and Projects: Why Participate?

OSS developers and projects	• Concentrate on developing great code without being distracted by legal and licensing issues • Adoption and visibility of projects, robust community management, forum for interaction • Indemnification of project committers
Multi-source Vendors	• A legal and licensing framework that makes it
Corporate IT	• A legal and licensing structure that simplifies corporate contributions to open source

Sponsorship Structure

Founding Sponsor: Microsoft ($1M per year)

- 2 appointed board seats
- Flexibility on number of galleries they can sponsor
- Project committers eligible to serve on Technical Advisory Board

Corporate Sponsor ($250,000 per year)

- Eligible to serve on Board of Directors
- Create and oversee one gallery
- Project committers eligible to serve on Technical Advisory Board

Gallery Sponsor ($150,000)

- Create and oversee one gallery

The Challenges of Starting From Scratch

- Cost
  - Legal, infrastructure, administrative charges add up quickly
    - First six months, including startup: $200,000 -$300,000
    - Ongoing operations: $20 -60K per month.
- Knowledge
  - Nonprofit governance, finances, and operations are non-intuitive
- Staffing
  - Finding the team of experienced open source and non-profit executives
  - Oversee executive management, finance, marketing, administrative, and technical functions requires staff
  - Finding the right OSS Legal, PR, and Marketing support
- Time to Market

Paula Hunter 10-12-02
This slide needs help!

Why Outercurve?

Outercurve Foundation

Experienced

Providing IP management,
development process,
and non-profit governance

Flexible

License, Technology,
and Forge Agnostic

Cost Effective

Providing experienced
Staffing and Operational
support at fractional cost

A Short Course on Standards

OUTERCURVE.ORG

Standards

- A standard is a specification that has been put through a consensus process by a collection of interested parties
- Interested parties might be a formal de jure process of by governments, an industry or trade organization with broad interests, or a consortium of narrow focus
- Standards exist to encourage and enable multiple implementations, regardless of the developing forum, and they benefit the customer
- A successful standard (economically) has many implementations; if there is only one implementation it is a failure (or worse a vendor specification)
- Standards practitioners are technology diplomats: your job is to expand your areas of economic influence and to defend sovereign territory.

Not Standards

- A vendor specification (regardless of whether it has been through a standards process or not) benefits the vendor by encouraging complements
- De facto technologies are NOT standards -- It refers to a single implementation with little consensus
- Corporate standards are ambiguous entities that may be selecting de facto products to reduce procurement problems, or may be pointing to real standards to encourage procurement choice

Standards and Existing Practice

- Good standards happen when a technology space matures to the point we can now see good abstractions
- If you try to standardize too early, you will fail against well engineered products
- Standards are based on "existing practice and experience", e.g.
  - POSIX/UNIX versus a myriad of proprietary operating system interfaces
  - ODF versus troff, runoff, GML, then SGML, etc.
  - TCP/IP versus SNA, DECnet, Banyan, etc.

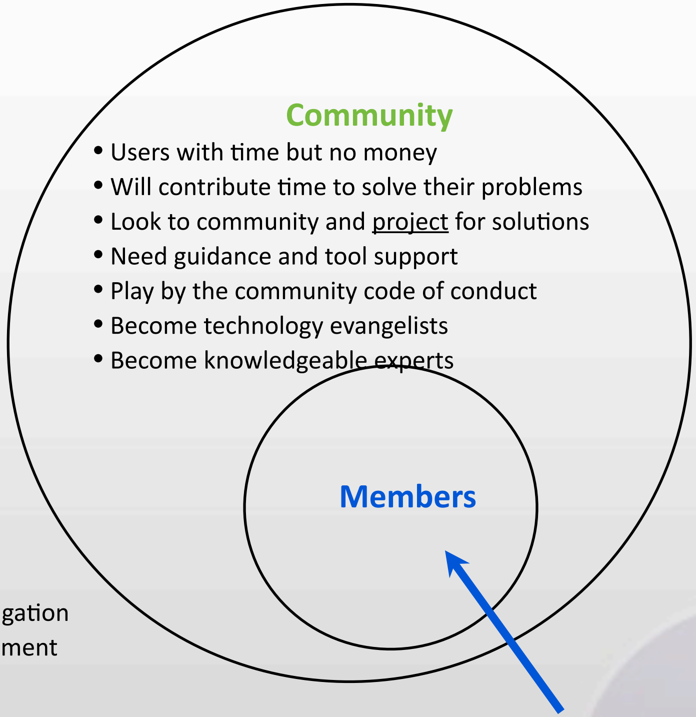
Standards and IP

- Patents and standards are opposite ends of the economic spectrum (Patents protect a single implementation of an idea.)
- Every standards development organization has an IPR policy
  - It is designed to give them early warning
  - No one wants to build a standard with a tax built in
  - Good IPR policies set expectations
- There is no way to protect against a troll from the outside

Standards and FOSS

- When an incumbent over-delivers in a market space, customers want "standards" so as to introduce competition: they signal this by complaining about price
- Competitors co-operate around an existing similar collaborative development project to form a standard to crack open the incumbent market position
- Essentially, the competitors attack the core revenue stream of the incumbent with a standard based on (but not identical to) collaborative technology they can share
- Collaborative technology == FOSS in today's market
- This means FOSS projects often conform to standards (rapidly) and lead quickly to implementations
- The incumbent still thinks its about delivering better innovation thereby exacerbating their own problems
- FOSS does NOT replace the need for standards — It fosters their creation

Understanding Community & Foundations



Community

- Users with time but no money
- Will contribute time to solve their problems
- Look to community and project for solutions
- Need guidance and tool support
- Play by the community code of conduct
- Become technology evangelists
- Become knowledgeable experts

Members

Foundations

- Provide IP management and risk mitigation
- Provide experienced project management
- Promote the technology (marketing)
- Provide education and expertise

Members

- Pay to manage the roadmap
- Share the cost of technology promotion
- Share the cost of clear IP management
- Play by the rules of membership

Examples of Standards and Collaboration

- The UNIX Wars were not about "UNIX"
  - DEC was the dominant mini-computer supplier and had over delivered on VAX/VMS
  - Competitors chose a technology base around which they had collaborated to create a standard
  - UNIX implementations quickly aligned to the standard
- ODF and Microsoft Office
  - Microsoft is the dominant productivity office suite and has over delivered with Microsoft Office
  - Sun, Novell, IBM chose a technology base around which they had collaborated to create a standard
  - OpenOffice.org quickly aligned to the standard, and implementations are appearing rapidly

OUTERCURVE.ORG