



OUTERcurve
FOUNDATION

Open Source, Business, and Mobile

Stephen R. Walli

Outercurve Foundation Technical Director

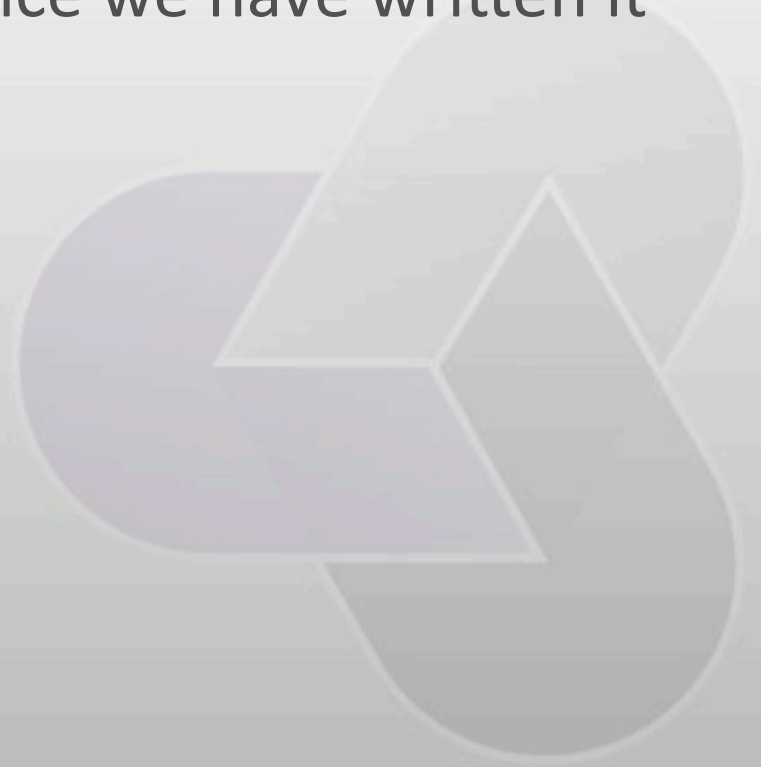
Twitter: @stephenwalli

30 March 2011

Free and Open Source Software



We have shared software since we have written it

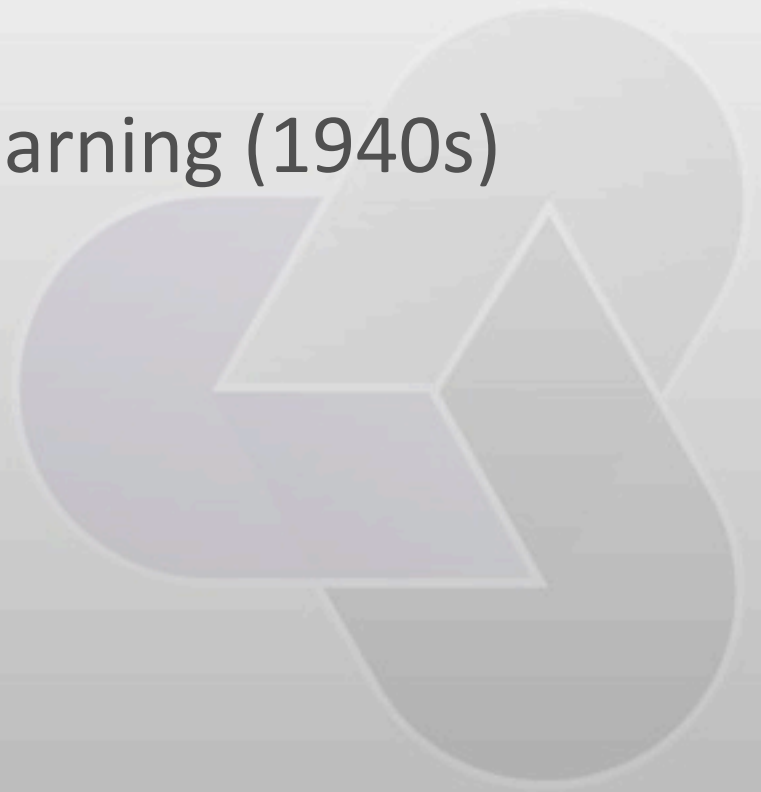


OUTERcurve.ORG

DECUS (1970s-80s)

SHARE (1950s-80s)

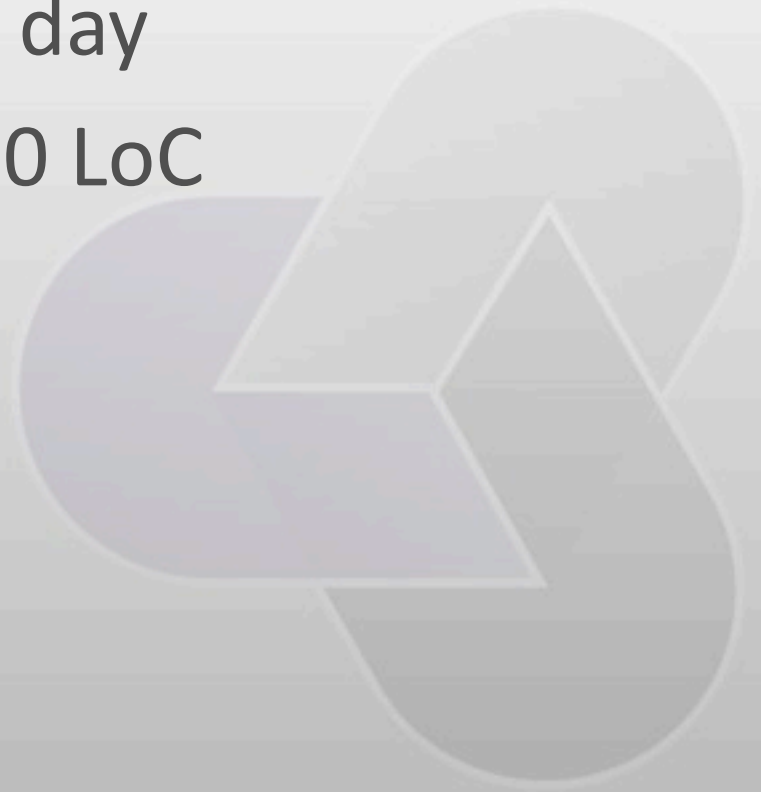
Institute for Advanced Learning (1940s)



Two Ratios

~20 LoC / day

1 bug : 1000 LoC

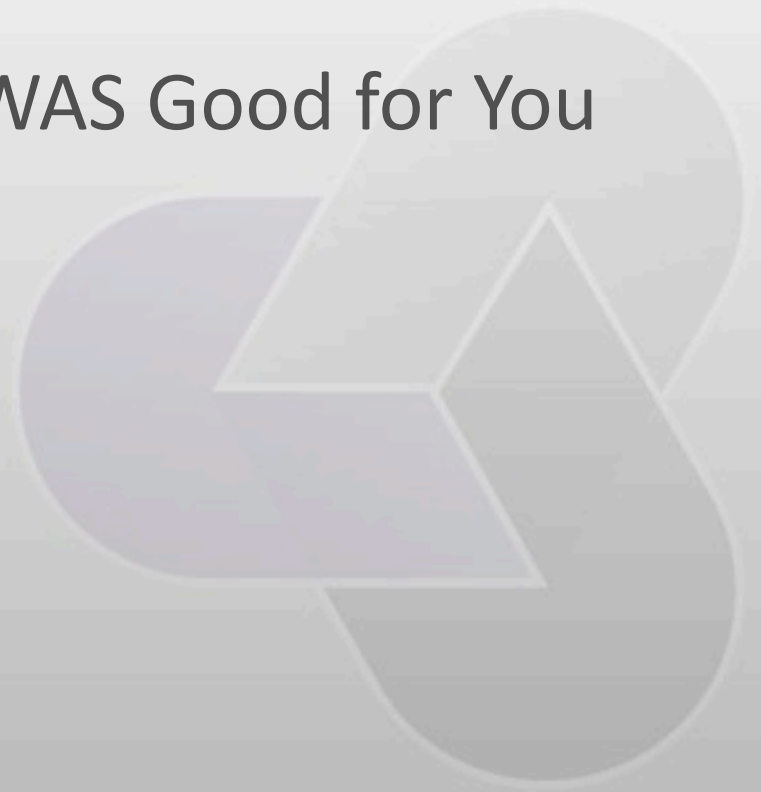


Plagiarism in software is good and indeed
necessary

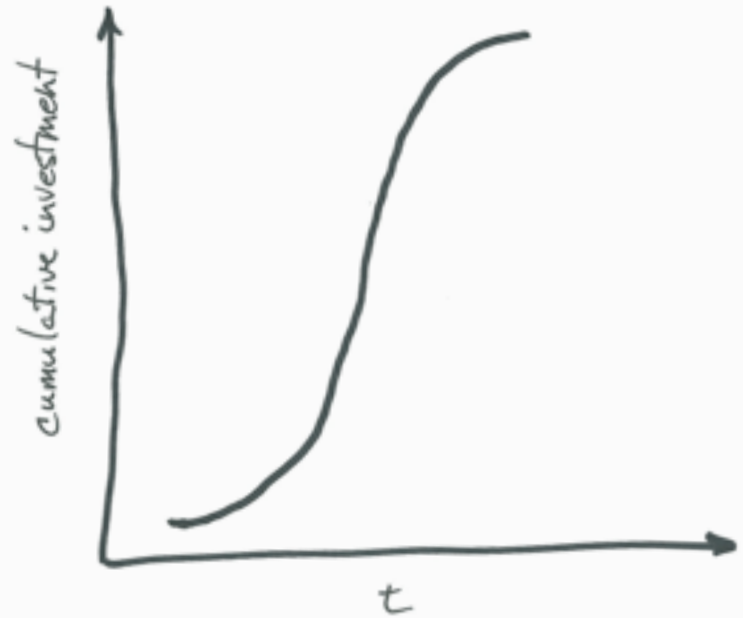
Two More Ratios

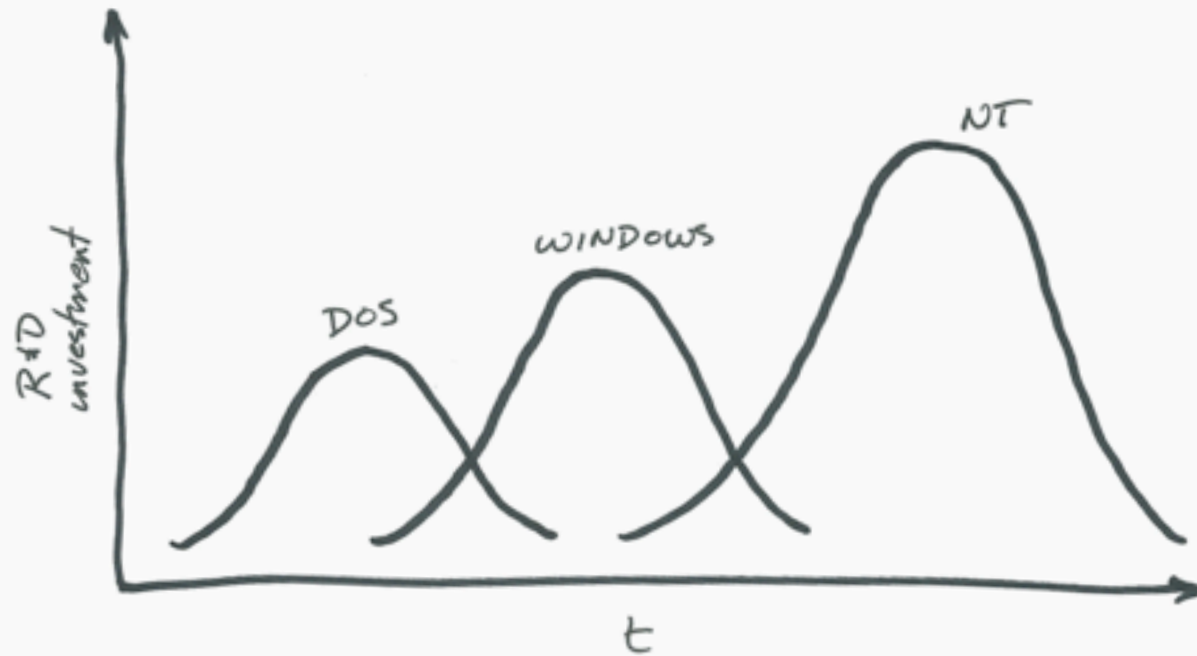
75% of an applications cost is “maintenance”
60% of applications never get deployed

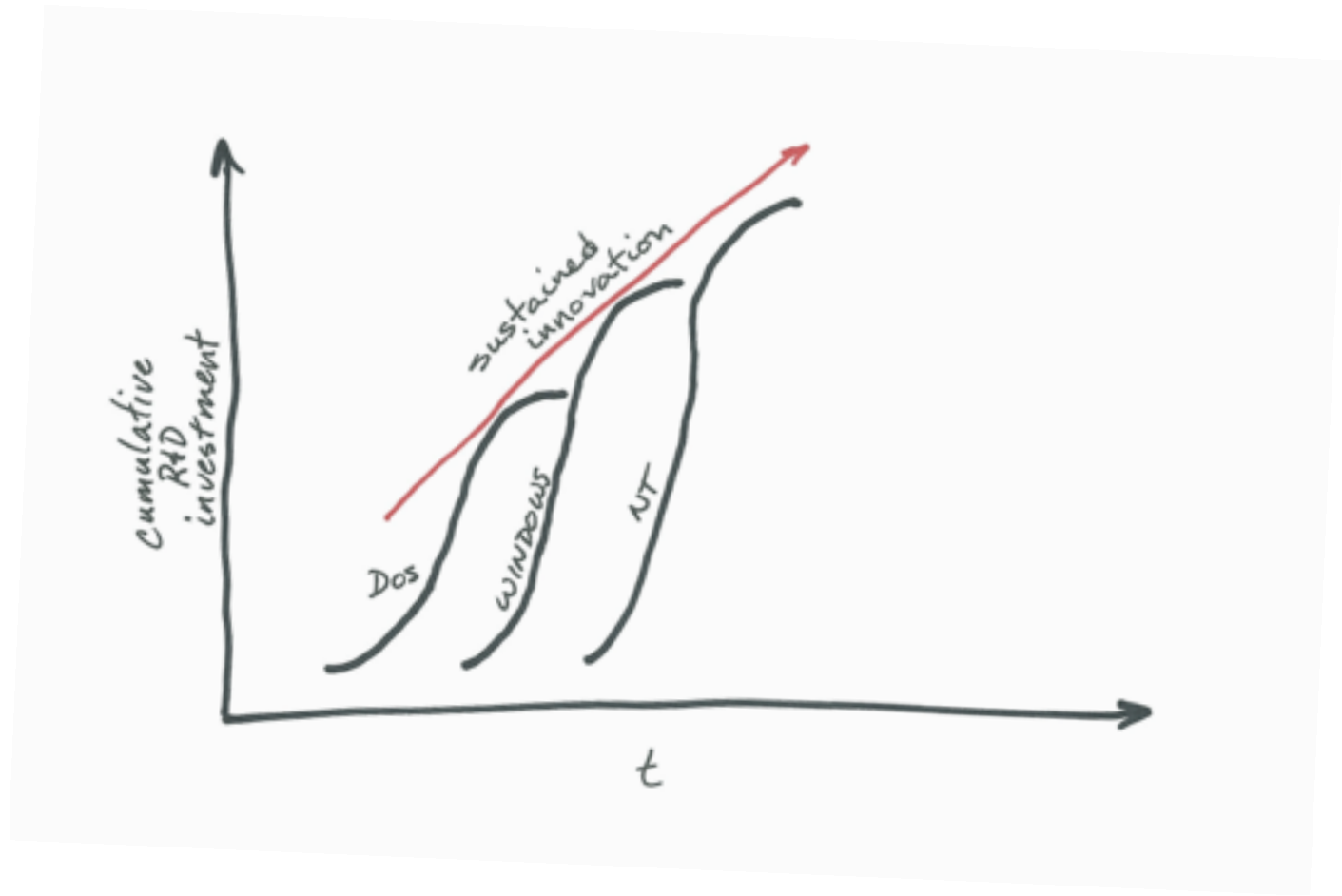
Proprietary Software WAS Good for You



OUTERcurve.ORG





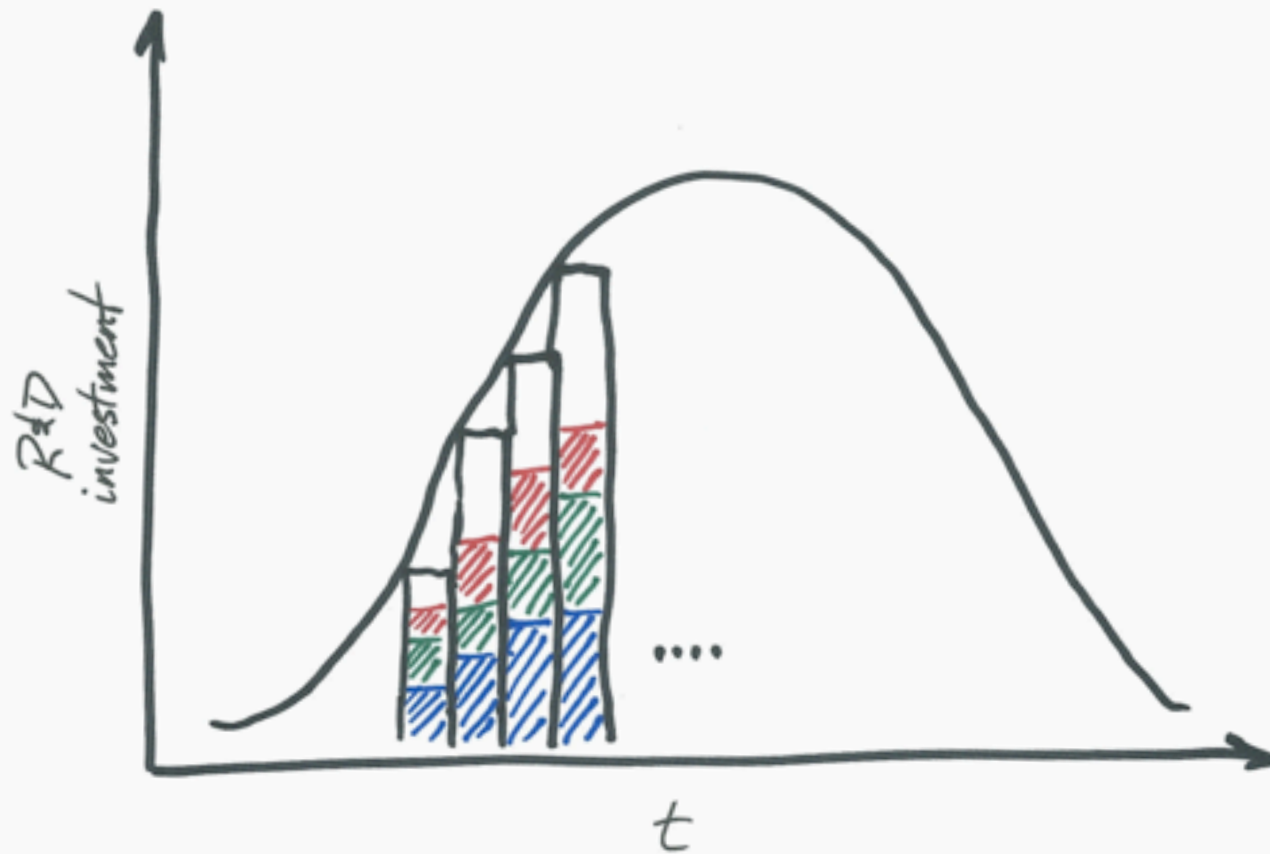


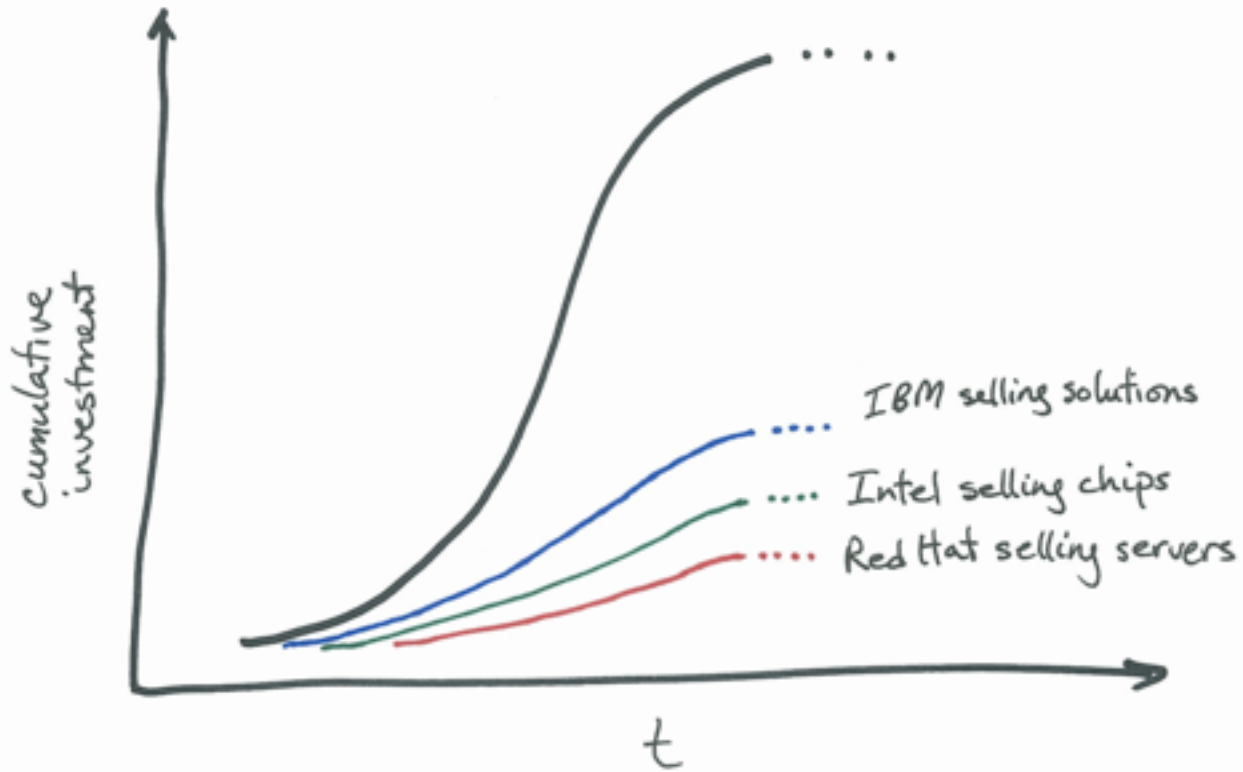
An then the Internet Happened ...

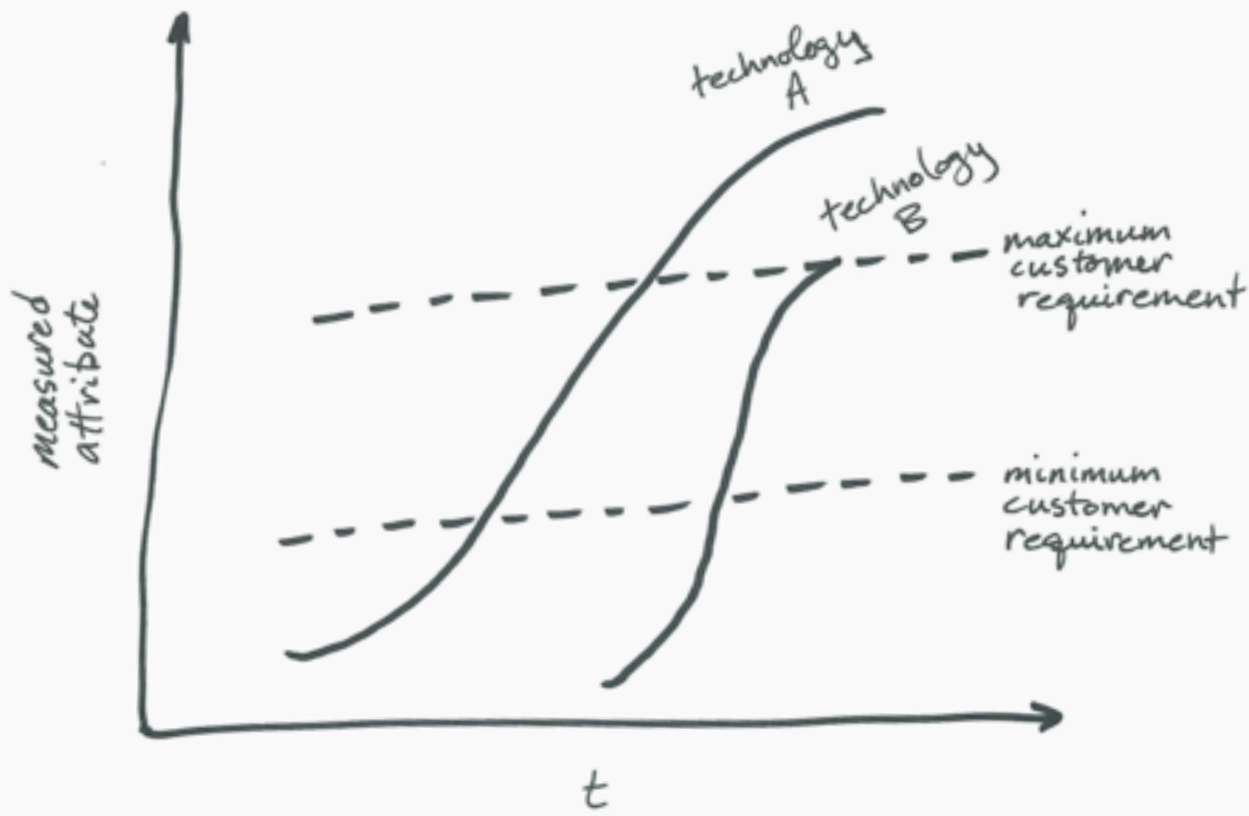
Tape-sized packets and conference latency
(or 300 baud modems)

versus

The Internet and the WWW



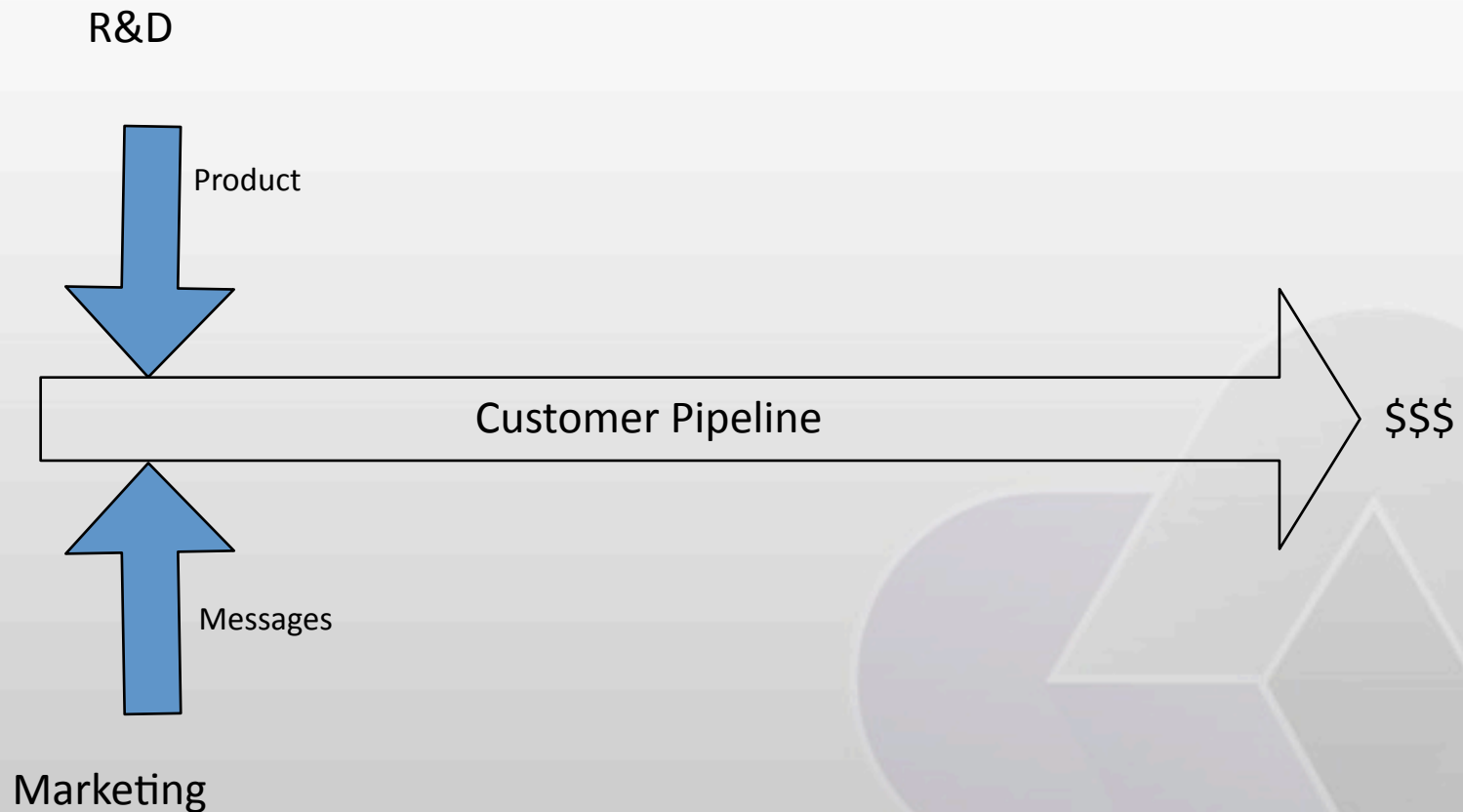




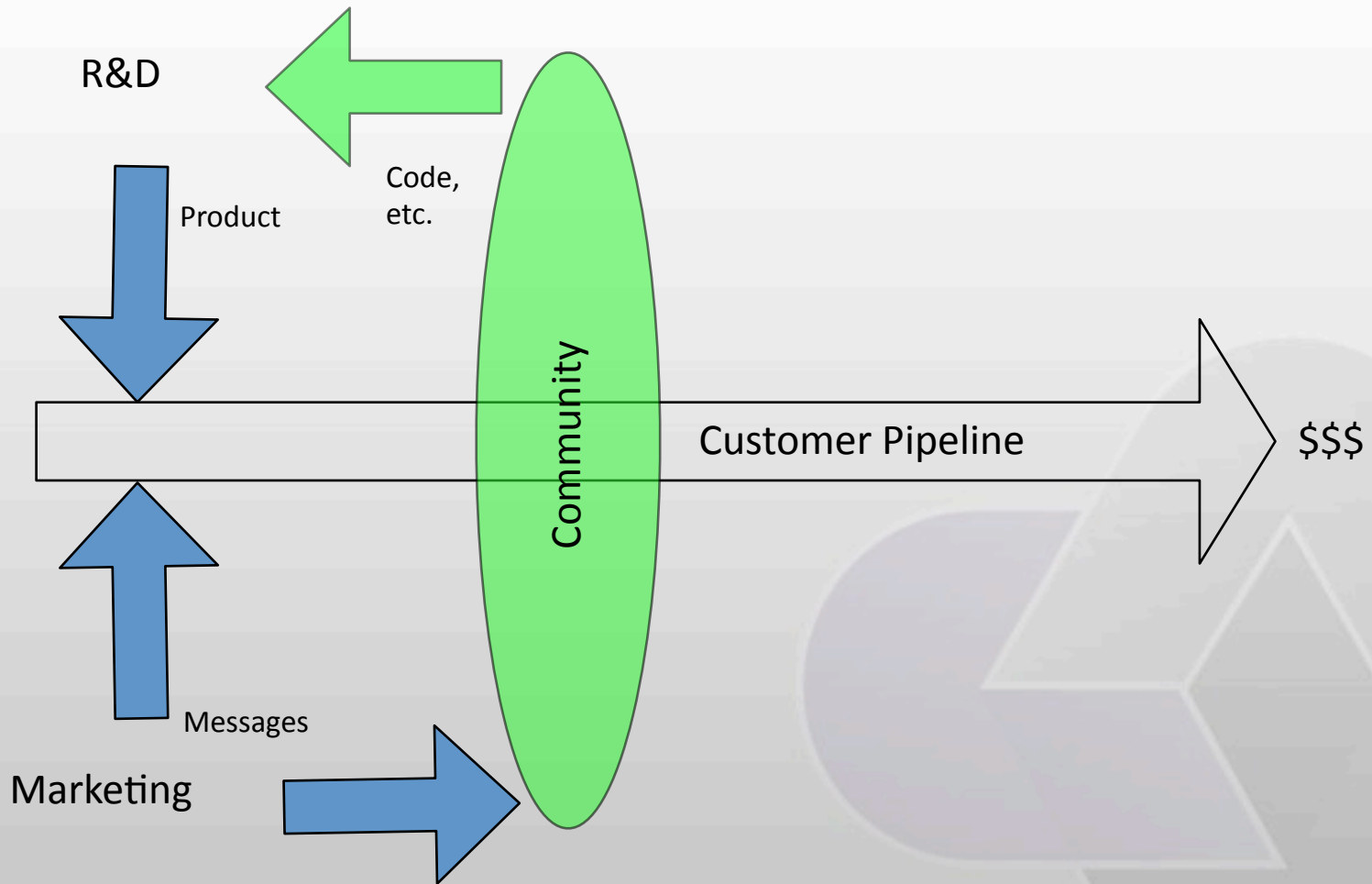
The Campfire Rule

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

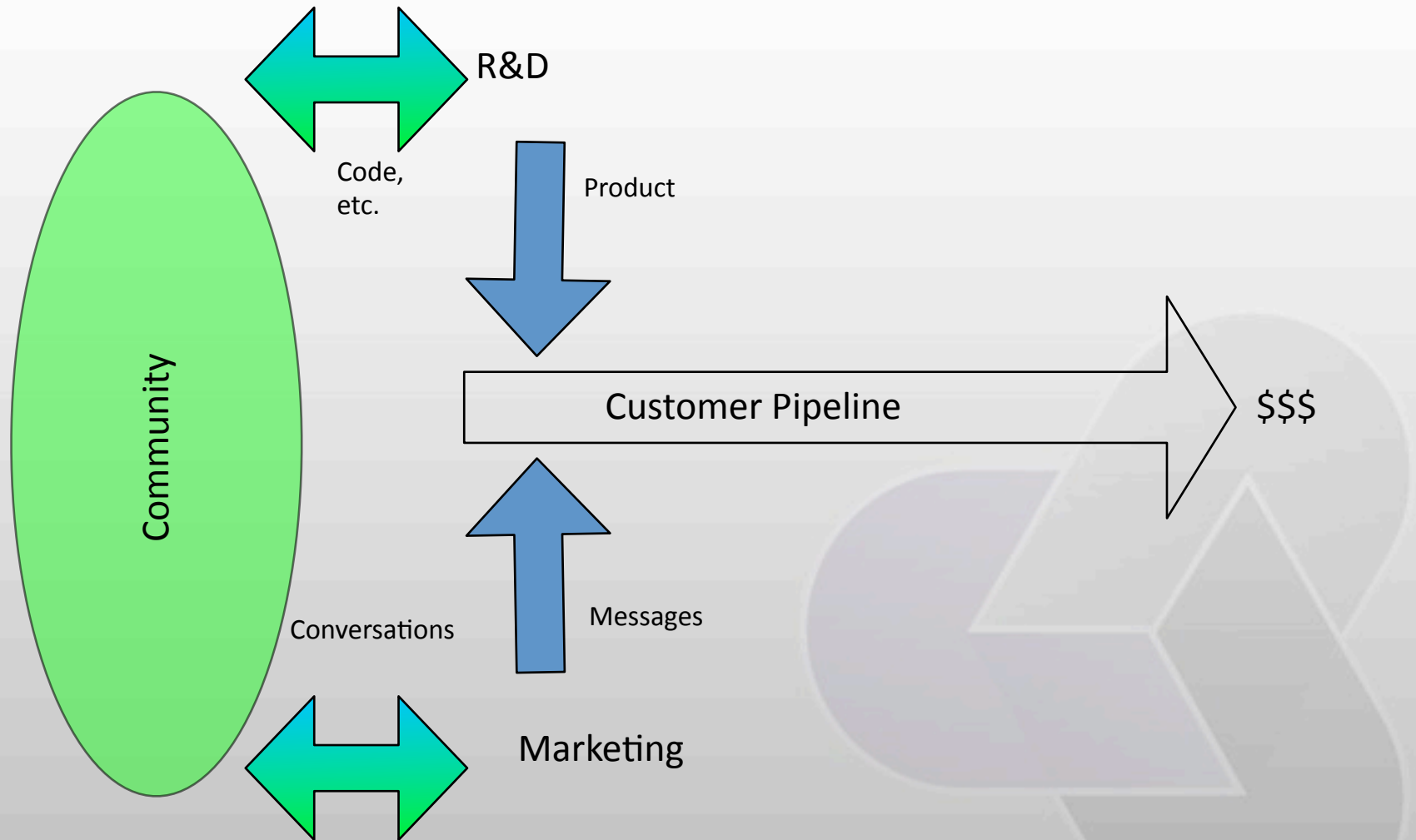
Traditional Software Business



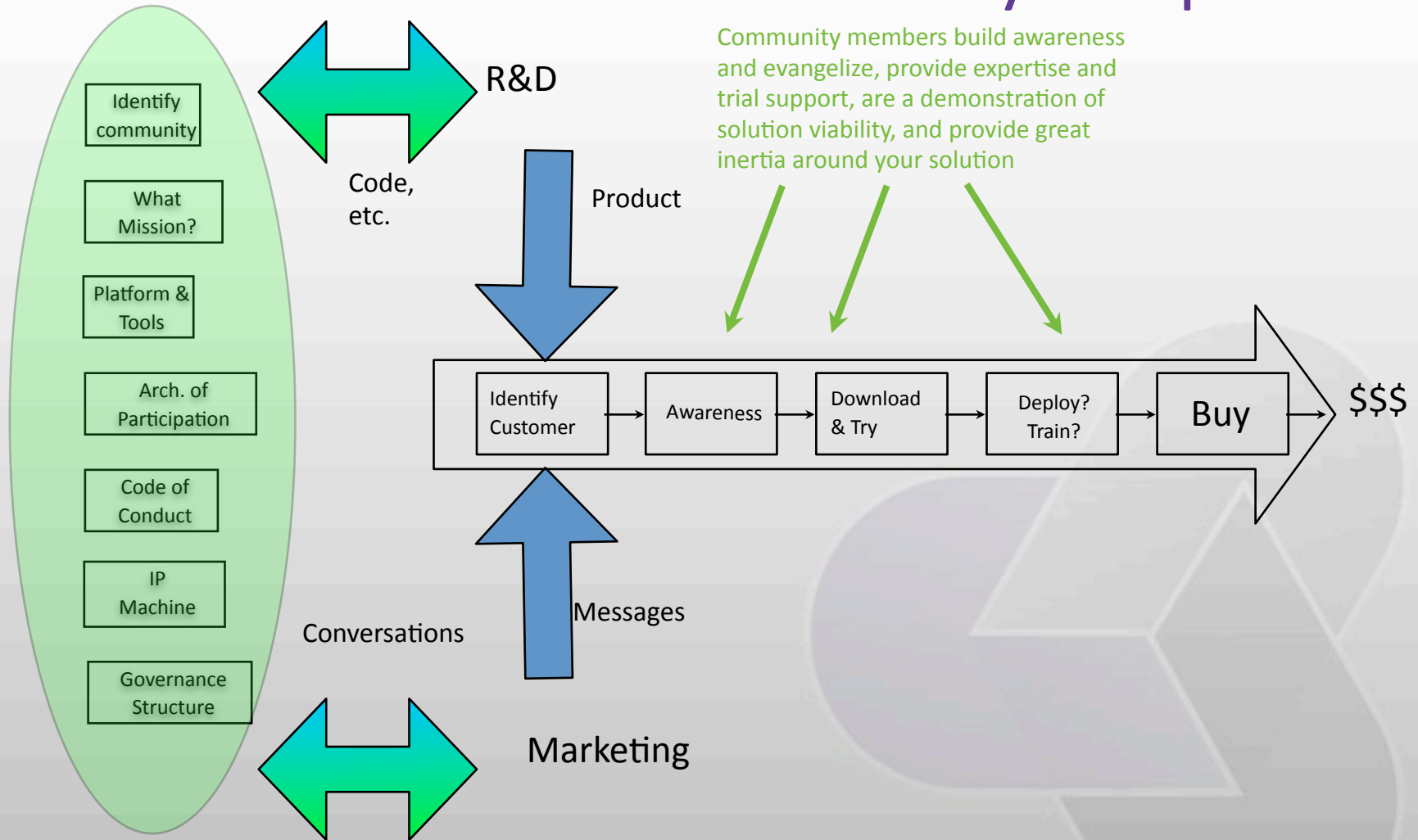
Misconceptions about Community



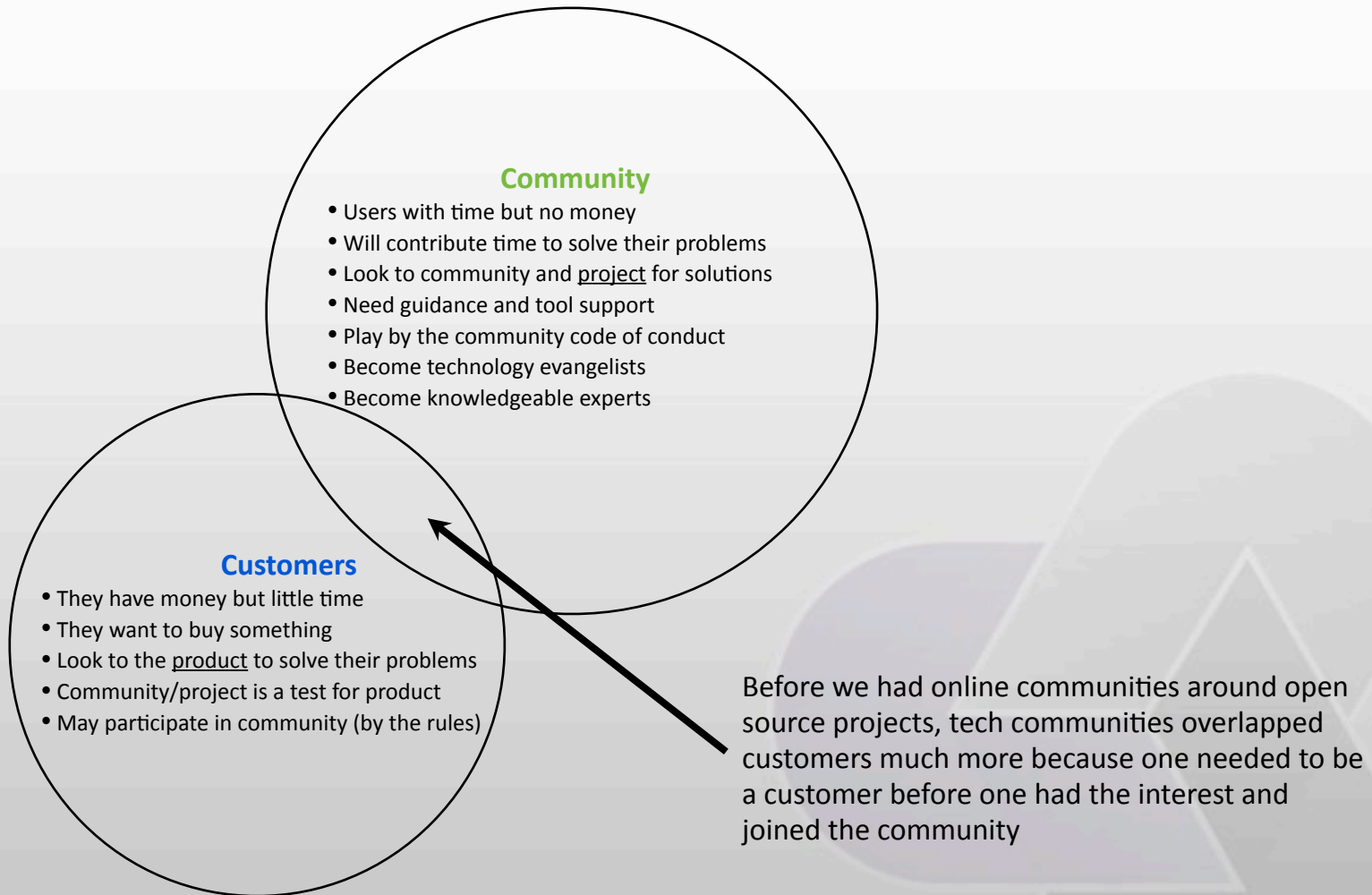
A Better Model for Community & Pipeline



A Better Model for Community & Pipeline



Understanding Community & Customers



Understanding Community and Open Source

Pure Open Source
Community

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
-

Traditional
Closed
Company

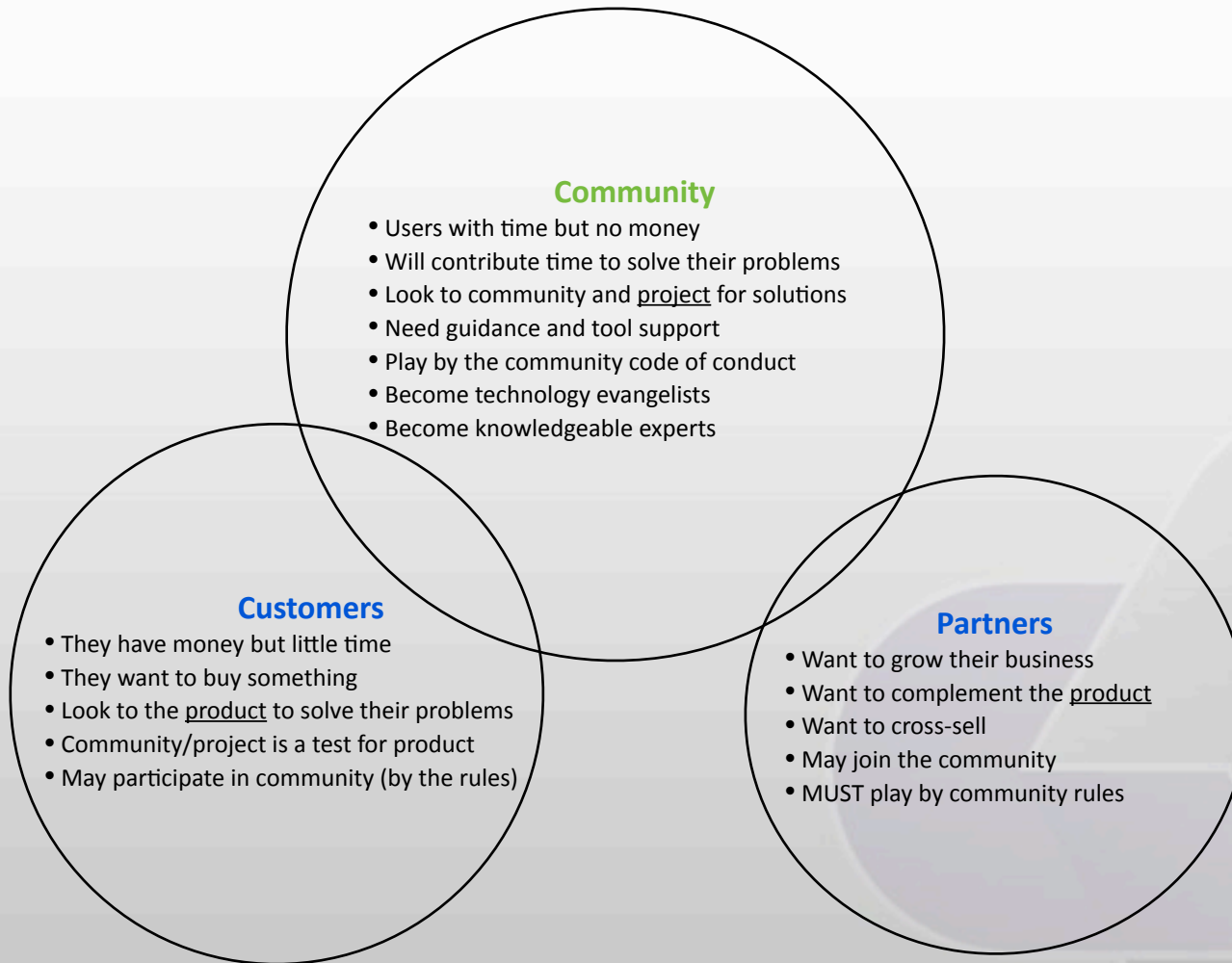
Customers

- 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



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”

“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)

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

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

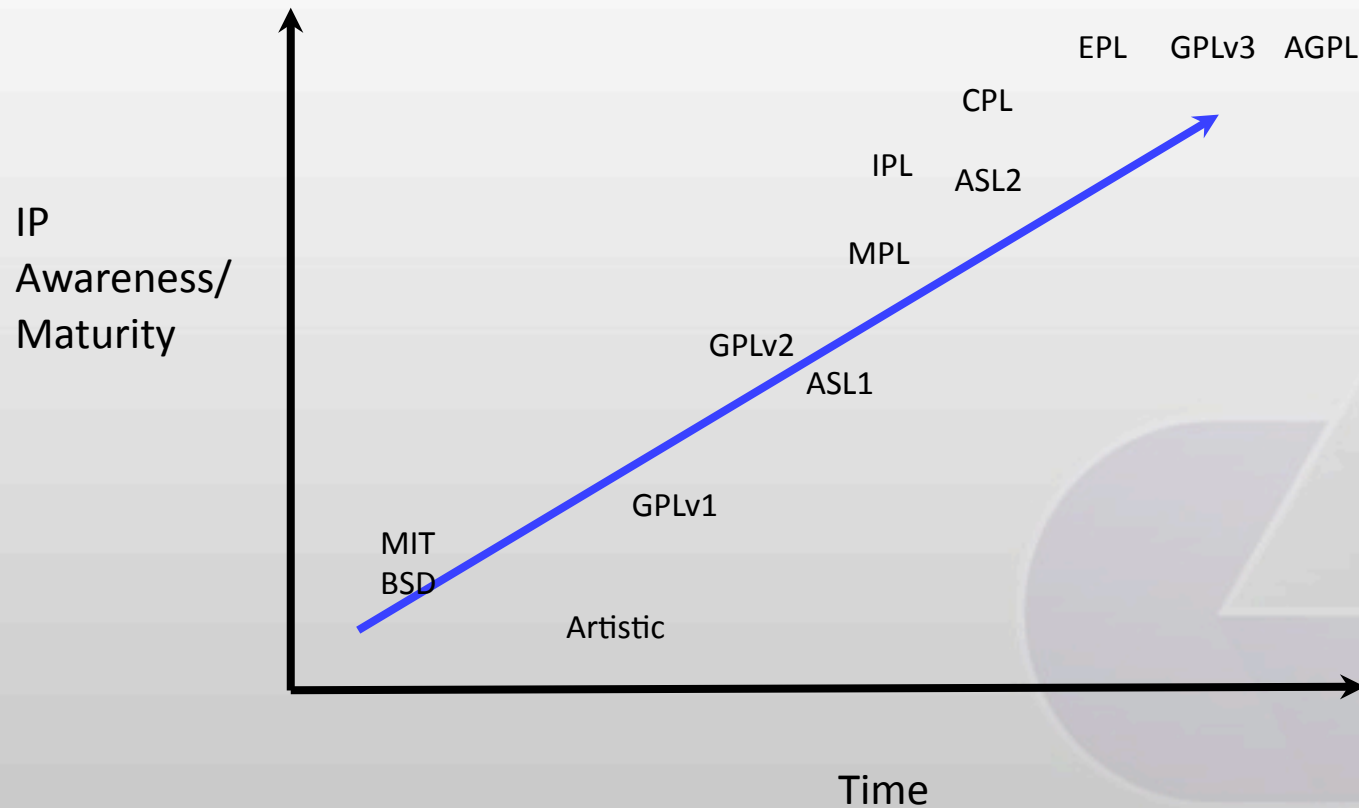
- Google

- 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 Maturation of OSS Licensing



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

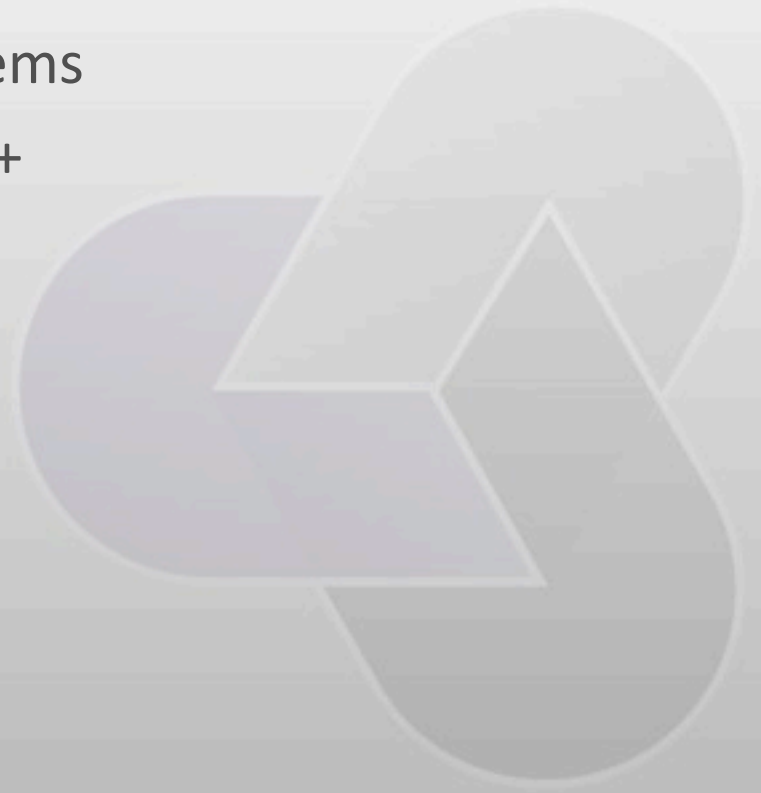
Some provide:

- 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

