

Open Source Software: Free Provision of a Complex Public Good

By James Bessen
Research on Innovation

•
•
•

Paradox of Free Software

- Std. proprietary products aggregate demand
 - Allow large investment
- Free goods: public externality
 - insufficient investment
- Why robust F/OS software?
 - E.g., Apache 64% of active web sites

-
-
-

Explanation: Complexity

- Standardized products cannot meet all complex needs
- F/OSS is NOT simple volunteerism
 - Exchange suited to complex goods

•
•
•

Complexity of Software

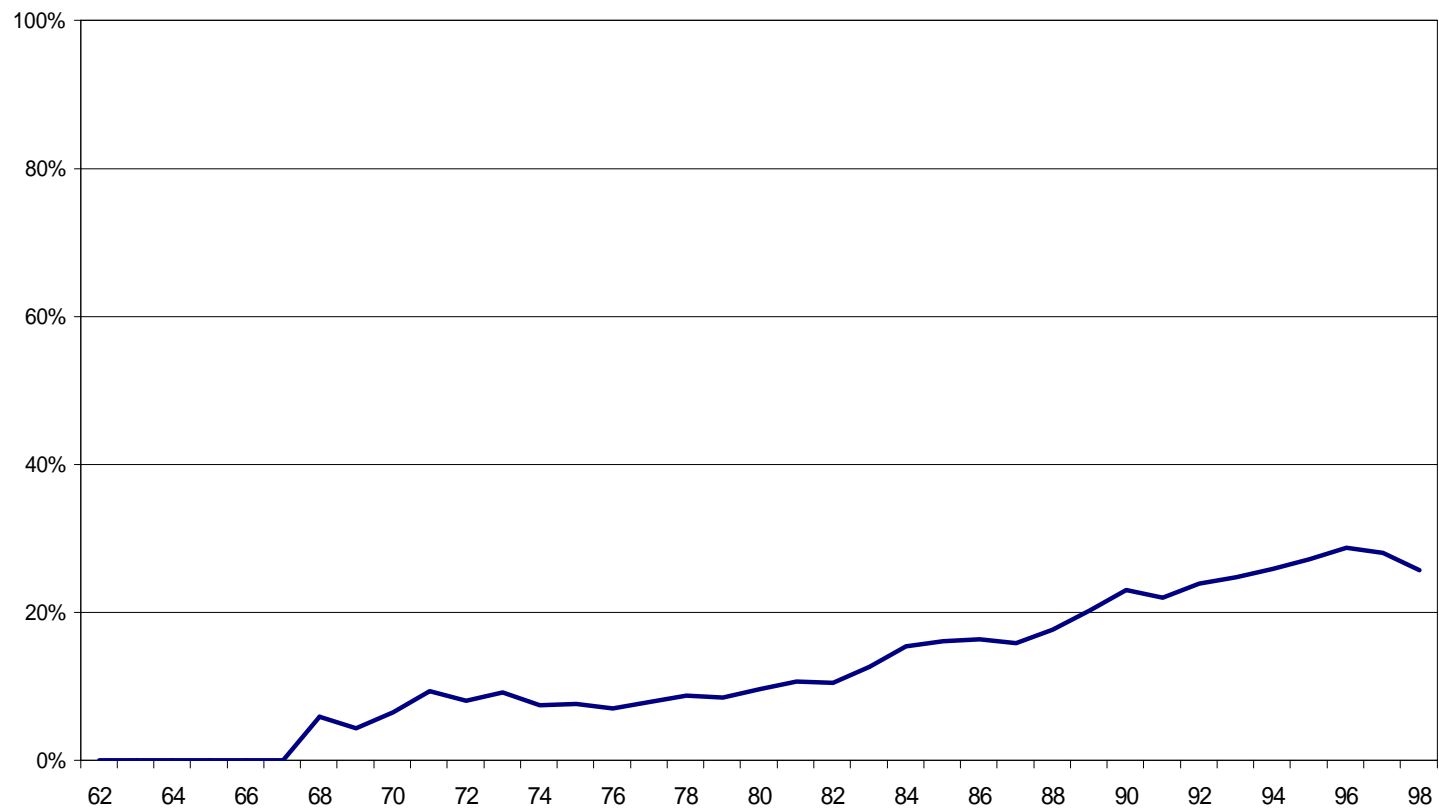
- Many features interact
- Consumers want particular “use-product” only
- Cost of supporting each use-product
- Customization: “soft” in software

•
•
•

Complexity limits standard SW

- Feature “bloat” and feature “wars”
- High debug and maintenance

Packaged SW Share



-
-
-

Proprietary customization

- Full custom usually infeasible
- Contract with std. producer
- Use API

•
•
•

But prop. custom limited, too

- Asymmetric information
- Incomplete contracts
- F/OSS provides a better way

-
-
-

“Free” Software

- Zero price
 - But total user cost is positive
 - NOT “freeware”
- Free to modify (libre)

•
•
•

Customization and F/OSS

- Apache security features
 - 19% users custom code
 - 33% use add-ons
- Represents large unmet demand

•
•
•

Individual modifications shared

- Apache first 3 years:
 - 6,092 features; 695 bug fixes
 - 300 add-ons
- Sharing part of F/OSS process
 - GPL
 - Other licenses similar in practice

-
-
-

Model

- Endogenous provision
 - Standard package
 - Custom API (& contract programming)
 - F/OSS
- Changes with complexity, market size
- Difference from literature
 - Explain participation of *firms*
 - Complex product spaces

-
-
-

Demand

- N consumers
- M possible features
- m features in use-product
- $w = P[\text{consumer will want a feature}] < 1/2$
- Expected market for use-product:

$$y(m) = w^m \cdot (1 - w)^{M-m}$$

-
-
-

Utility

- u = utility from base product
- v = utility from each feature used
- p = price

$$u + m \cdot v_i - p$$

-
-
-

Cost

- c = cost of developing base product
- d = cost of “debugging” a use-product
 - includes enhancement
 - linear
- Total cost for all use-products debugged:

$$c + 2^M d$$

•
•
•

Proprietary provision

- Standard package
 - price discrimination of versions:

$$p_m = m v + u$$

- API
 - single price, p_{API}

•
•
•

F/OSS provision

- Stage 1: base product

- with n expected developers

$$u + m v \geq E[c/n] + d$$

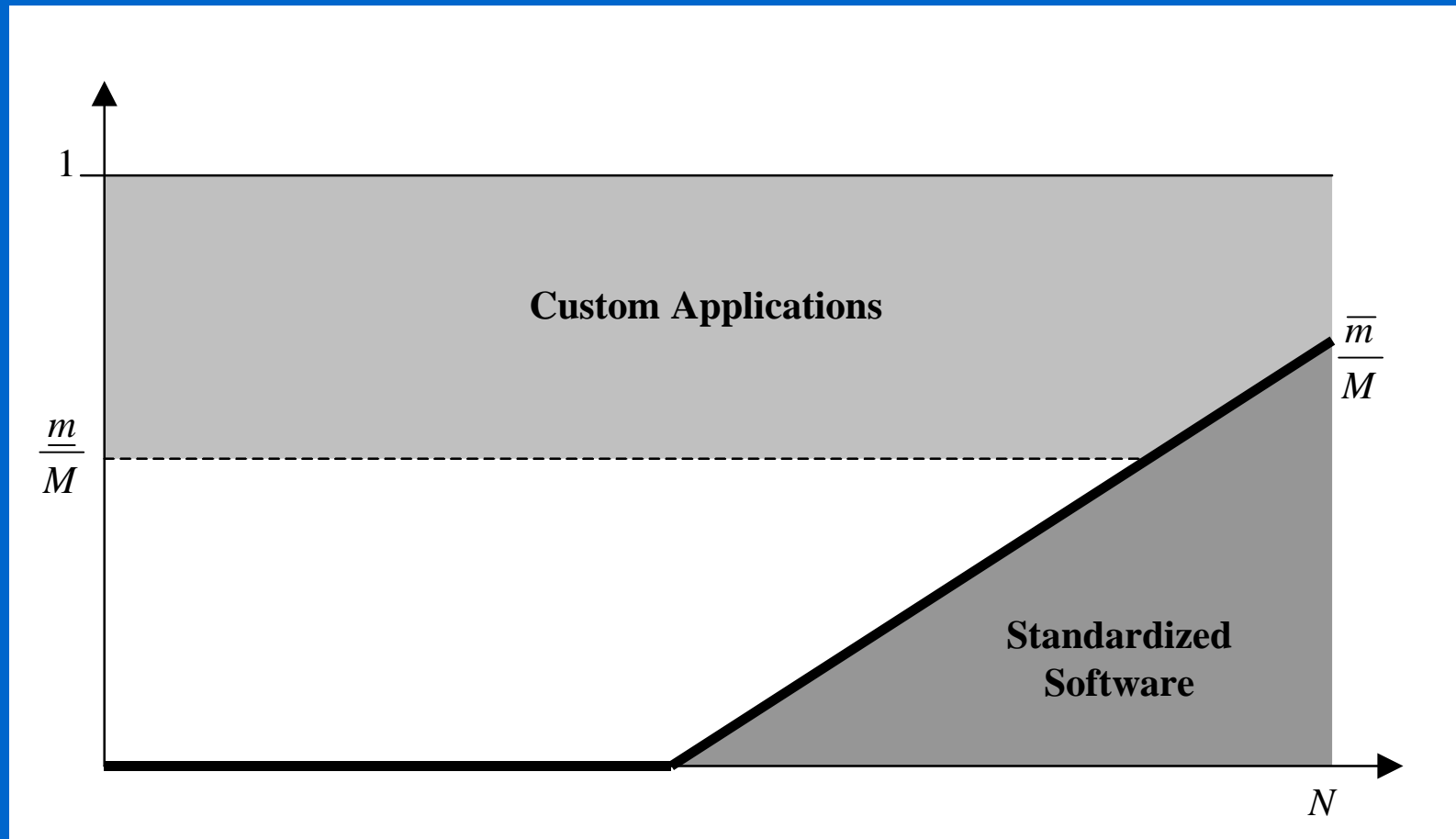
- Free riding

- Individual motivation

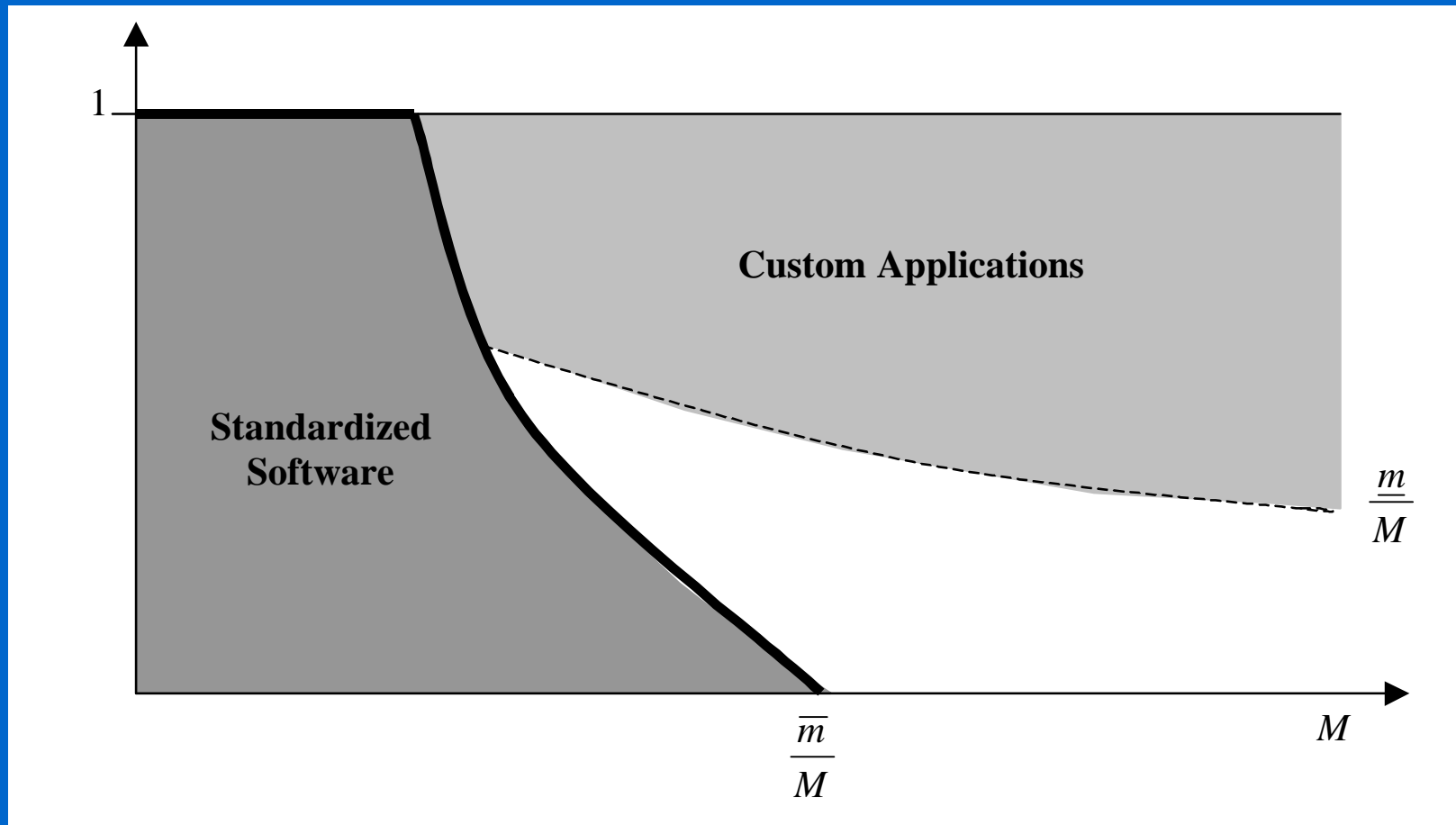
- Stage 2: individual enhancements

$$u + m v \geq d$$

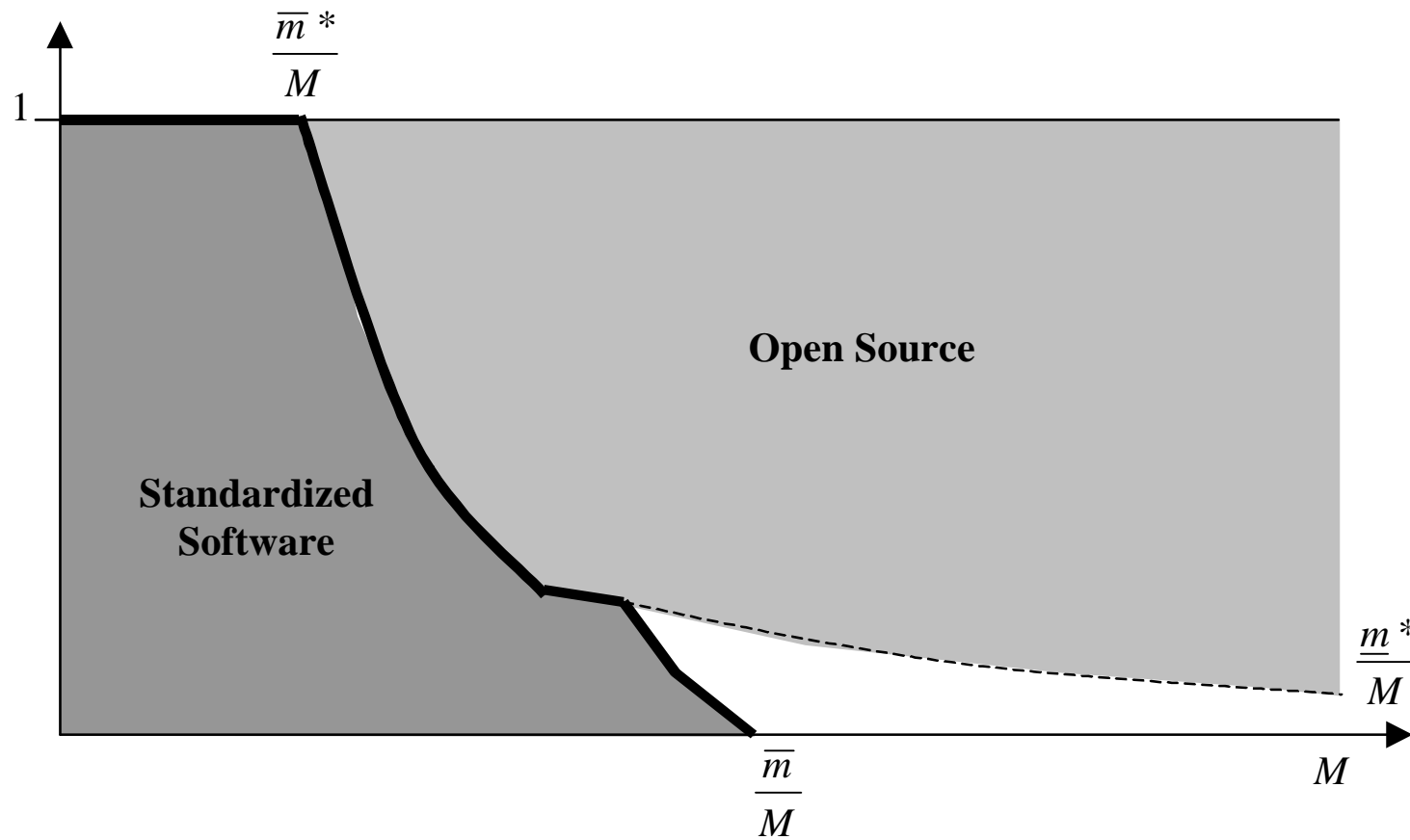
Proprietary provision vs. N



Proprietary provision vs. M



Combined provision



-
-
-

Summary

- Standardized SW: large markets, simple needs
- F/OSS: Complex markets, needs
- F/OSS dominates proprietary custom
 - price discrimination assumption
- F/OSS increases overall provision, welfare

-
-
-

Conclusion

- F/OSS complements standard SW
 - Standard SW aggregates demand
 - F/OSS aggregates *supply*
- F/OSS an *extension* of market
 - software exchanged for promise of enhancements