

Chapter 3 The Software Industry

9534801
資管博一 李易霖

© 2007 Institute of Information Management

National Chiao Tung University

Agenda

- Principles of Software Production
- The Determination of Software Variety
- Software Variety Under Hardware Competition
 - Incompatible systems
 - Compatible systems
- Software Variety and Partial Competition
 - Determination of brand-specific software variety
 - The effects of changing the degree of compatibility
- Software Piracy
- Software Pricing and Market Segmentation
- Empirical Findings

© 2007 Institute of Information Management

National Chiao Tung University

Principles of Software Production

- Software packages are produced by software firms that are generally independent of hardware producers
- Software packages are regarded as *supporting services* for the hardware
- In this analysis, instead of network externalities, we assume that there are no externalities
- Consumers determine the value of the hardware machine by the variety of software supporting the machine in addition to hardware's price

© 2007 Institute of Information Management

National Chiao Tung University

Principles of Software Production

- Hardware machines are said to be compatible if they can run the same software. In this case, we can say that the machines are software compatible.

© 2007 Institute of Information Management

National Chiao Tung University

Principles of Software Production

- Symbols and Meanings

Symbol	Meaning
q	Number of buyers
ϕ	Cost of developing, testing and debugging software
μ	Cost of shipping one piece of software to a consumer
$TC(q)$	total cost function
$AC(q)$	average cost as total production per unit function
$MC(q)$	marginal cost
η	Customers who purchase a certain computer brand
ω	Total income of each consumer allocated to be spent on on computer
p	consumer pay for purchase hardware
e	$\omega - p$
s	Number of software packages written for this machine
ϕ	Fixed cost of developing one software package

© 2007 Institute of Information Management

National Chiao Tung University

Principles of Software Production

- General Definition

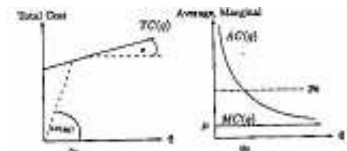
$$AC(q) \stackrel{\text{def}}{=} \frac{TC(q)}{q}$$

$$MC(q) \stackrel{\text{def}}{=} \frac{\Delta TC(q)}{\Delta q}$$

- Software Industry

$$AC(q) = \frac{\phi}{q} + \mu$$

$$MC(q) = \mu$$



© 2007 Institute of Information Management

National Chiao Tung University

Principles of Software Production

- **For the software production cost**
 - a) At every pricing, p_0 , there exists a minimum sales q_0
if sales $q > q_0$, positive profit
if sales $q < q_0$, profit loss
 - b) Cost-based pricing is not appropriate for software

The Determination of Software Variety

- **Assumption:**
The variety (number) of software packages equals total consumer expenditure on this software divided by the software development cost

$$S = \frac{\eta e}{\phi} = \frac{\eta(\omega - p)}{\phi}$$

- A reduction in hardware price increases the amount consumers spend on software, thereby increase the variety of software available for the machine

The Determination of Software Variety

- Utility function of using computer

$$U = \begin{cases} \alpha s - p, & \text{Buys the computer and entire variety of software} \\ 0, & \text{Buys none} \end{cases}$$
- Now a monopoly hardware producer who set its price p to maximum profit. In view of the above function, it observes s and set the maximum prices each consumer is willing to pay, which $p^m = \alpha s$

The Determination of Software Variety

- Substituting $p^m = \alpha s$ into utility function

$$p^m = \frac{\alpha \eta (\omega - p^m)}{\phi}, \text{ hence } p^m = \frac{\alpha \eta \omega}{\alpha \eta + \phi}$$
- **The price charged by a monopoly hardware firm**
 - a) *increases with consumer love for software variety parameter, α*
consumer income, ω
number of consumers buying the machine, η
 - b) *decreases with software development cost, ϕ*

The Determination of Software Variety

- A higher utility from software variety enhances the value of the hardware machine, thereby enabling the hardware seller to raise its price
- An increase in income results in a higher demand and therefore a higher monopoly price
- When more people buy the machine, more software packages are produced, hence increasing price

The Determination of Software Variety

- Substituting $e = \omega - p$

$$s = \frac{\eta \omega}{\alpha \eta + \phi} \text{ and } e = \frac{\phi \omega}{\alpha \eta + \phi}$$
- **Under a monopoly hardware seller**
 - (a) *Equilibrium software variety, s , increases with number of consumers η , consumer income ω*
decreases with consumer love for software variety parameter, α
software development cost, ϕ
 - (b) *Consumer's software expenditure, e , increases with consumer income ω , software development cost ϕ*
decreases with consumer love for software variety parameter α
number of consumers, η

Software Variety Under Hardware Competition

- A duopoly computer hardware industry competition for consumers whose utility increases with the variety of software available for each brand.
- s_A : amount of software available for brand A
- s_B : amount of software available for brand B
- q_A : the number of A-Users
- q_B : the number of B-Users

$$s_i \stackrel{\text{def}}{=} \frac{q_i e_i}{\phi} = \frac{q_i (\omega - p_i)}{\phi}$$

where $e_i = \omega - p_i$ measures a consumer's total expenditure on software.

© 2007 Institute of Information Management

National Chiao Tung University

Software Variety Under Hardware Competition

- The variety of software packages supporting each brand equals total number of consumers using the hardware brand divided by the software development cost.**

$$s_i \stackrel{\text{def}}{=} \frac{q_i}{\phi}, i = A, B$$

- Utility function of each type i consumer is given by

$$U_i = \begin{cases} \alpha s_i - p_i & , \text{Buys brand } i, i \text{ is incompatible with } j \\ \alpha s_j - p_j - \delta & , \text{Buys brand } j, j \text{ is incompatible with } i \\ \alpha(s_A + s_B) - p_i & , \text{Buys brand } i, i \text{ is compatible with } j \\ \alpha(s_A + s_B) - p_j - \delta & , \text{Buys brand } j, j \text{ is compatible with } i \end{cases}$$

δ : switching costs associated with having consumer type i learning to operate computer brand j

© 2007 Institute of Information Management

National Chiao Tung University

Software Variety Under Hardware Competition Incompatible systems

- Brand i producer undercuts brand j producer, by setting p'_i sufficiently low to satisfy**

$$p'_i \leq p_j - \delta + \alpha(s'_i - s_j) = p_j - \delta + \alpha\left(\frac{2\eta}{\phi} - \frac{\eta}{\phi}\right) = p_j - \delta + \frac{\alpha\eta}{\phi}$$

© 2007 Institute of Information Management

National Chiao Tung University

Software Variety Under Hardware Competition Incompatible systems

- UPE (Undercut-Proof Equilibrium)**
 - A pair of hardware price constitutes an Undercut-Proof Equilibrium if

$$\pi_B = p_B^U \eta \geq \left(p_A - \delta + \frac{\alpha\eta}{\phi}\right) 2\eta$$

$$\pi_A = p_A^U \eta \geq \left(p_B - \delta + \frac{\alpha\eta}{\phi}\right) 2\eta$$

$$p_A^I = p_B^I = \frac{2(\phi\delta - \alpha\eta)}{\phi}, \text{ and } \pi_A^I = \pi_B^I = \frac{2\eta(\phi\delta - \alpha\eta)}{\phi}$$

© 2007 Institute of Information Management

National Chiao Tung University

Software Variety Under Hardware Competition Incompatible systems

- An increase in consumers' preference for variety of software reduces both hardware prices and profits. Formally, as α increases, p_A , p_B , π_A , π_B and all decrease.**
- When the machines are incompatible**
 - An increase in consumer population decrease firms' profits when consumers place a sufficient value on software variety
 - Both prices and profits increase with the differential parameter, δ decrease with the software development cost, ϕ

© 2007 Institute of Information Management

National Chiao Tung University

Software Variety Under Hardware Competition Compatible systems

- Brand i producer undercuts brand j producer, by setting p'_i sufficiently low to satisfy

$$p'_i \leq p_j - \delta + \alpha(s'_i + s'_B - s_A - s_B) = p_j - \delta$$
- UPE (Undercut-Proof Equilibrium)**
 - Must satisfy $\eta p_i^U \geq 2\eta(p_j - \delta)$

$$\eta p = 2\eta(p - \delta) \text{ or } p^C = 2\delta$$
- Equilibrium duopoly hardware prices and profits are higher when the machines are compatible than they are incompatible**

© 2007 Institute of Information Management

National Chiao Tung University

Software Variety and Partial Compatibility

- A computer brand i is said to be partially compatible with a ρ_i degree of compatibility with computer brand j , if a fraction ρ_i of the total software written specifically for brand j can also be run on computer brand i .

© 2007 Institute of Information Management

National Chiao Tung University

Software Variety and Partial Compatibility

- S_i is the number of software package written specifically for i machine(operation system), $i=A,B$ respectively. Then, the effective number of software packages available to A and B users are:

$$S_A^{\text{def}} = S_A + \rho_A S_B \text{ and } S_B^{\text{def}} = S_B + \rho_B S_A$$

$$U^{\text{def}} = \begin{cases} S_A & \text{buys computer brand A} \\ S_B & \text{buys computer brand B} \end{cases}$$

© 2007 Institute of Information Management

National Chiao Tung University

Software Variety and Partial Compatibility

- There are η_p programmers who can each produce $\gamma > 0$ software packages either for machine A or machine B but not both.
- The total number of packages produced is:

$$S_A + S_B = \gamma \eta_p$$

© 2007 Institute of Information Management

National Chiao Tung University

Software Variety and Partial Compatibility determination of brand-specific software variety

$$S_A = S_A + \rho_A S_B = S_B + \rho_B S_A = S_B$$

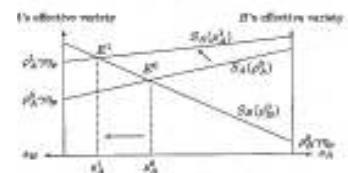
$$\frac{S_A}{S_B} = \frac{1 - \rho_A}{1 - \rho_B}$$

$$S_A = \frac{\gamma \eta_p (1 - \rho_A)}{(2 - \rho_A - \rho_B)}$$

$$S_B = \frac{\gamma \eta_p (1 - \rho_B)}{(2 - \rho_A - \rho_B)}$$

$$S_A(\rho_A) = S_A + \rho_A (\gamma \eta_p - S_A) = \rho_A \gamma \eta_p + (1 - \rho_A) S_A$$

$$S_B(\rho_B) = S_B + \rho_B (\gamma \eta_p - S_B) = \rho_B \gamma \eta_p + (1 - \rho_B) S_B$$



© 2007 Institute of Information Management

National Chiao Tung University

Software Variety and Partial Compatibility

The effects of changing the degree of compatibility on the variety of brand-specific software

- When there are two software industries, each producing brand-specific software, an increase in the degree of compatibility of the A-machine with the software written for the B-machine
 - (a) will reduce the variety of software specifically written for the A machine
 - (b) will increase the variety of software specifically written for the B machine

© 2007 Institute of Information Management

National Chiao Tung University

Software Piracy

- Two reasons for removing protection against copying
 - Consumers were annoyed by the consequences protective devices had
 - When the market expands and competition intensifies, firms have strategic incentives to remove protection or increase the number of consumers using their packages

© 2007 Institute of Information Management

National Chiao Tung University

Software Piracy

- Consumers are divided according to
 - Support-oriented consumer (type O) : who gain an extra utility from service and support provide by the software firm to their legal customers. We assume that there are η support-oriented potential consumers.
 - Support-oriented consumer (type I) : who don't derive utility from the service and support provided by the software firms to their legal customers. We assume that there are η support-independent potential consumers.
 - Total population = 2η

© 2007 Institute of Information Management

National Chiao Tung University

Software Piracy

- Here we assume**
 - The software firm bundles the support with purchase.**
 - Illegal software users cannot obtain support from an independent supplier.**

© 2007 Institute of Information Management

National Chiao Tung University

Software Piracy

- The utility of support-oriented consumer is

$$U^O = \begin{cases} (1+\sigma)q - p & , \text{buys the software} \\ q & , \text{pirates (steals) the software} \\ 0 & , \text{doesn't use this software} \end{cases}$$
- The utility of support-independent consumer is

$$U^I = \begin{cases} q - p & , \text{buys the software} \\ q & , \text{pirates (steals) the software} \\ 0 & , \text{doesn't use this software} \end{cases}$$

© 2007 Institute of Information Management

National Chiao Tung University

Software Piracy

- Support-oriented consumers would prefer buying software over pirating software if $p \leq \sigma q$; i.e., if the price of the software package doesn't exceed the values of service provide by the software firm to its legal users.**
- If software isn't copy protected, support-independent consumers never buy software; i.e., they either pirate it or don't use it at all.**

© 2007 Institute of Information Management

National Chiao Tung University

Software Piracy

- Monopoly software firm has two options in choosing its protection policy:**
 - (a) Nonprotection policy (n): Any consumer can costlessly pirate the software, but can't obtain any service from the software firm.**
 - (b) Protection policy (p): Installing devices and/or implementing an enforcement policy in order to make software piracy practically impossible.**

© 2007 Institute of Information Management

National Chiao Tung University

Software Piracy

- No copy protection

$$p^n = \sigma 2\eta$$

$$\pi^n = p^n \eta = 2\sigma \eta^2$$
- Copy protection

$$p^{p,H} = (1+\sigma)\eta$$

$$\pi^{p,H} = (1+\sigma)\eta^2$$

$$p^{p,L} = 2\eta$$

$$\pi^{p,L} = 4\eta^2$$

$$\pi^{p,H} \geq \pi^{p,L} \Leftrightarrow \sigma \geq 3$$

$$p^p = \begin{cases} (1+\sigma)\eta & , \text{if } \delta \geq 3 \\ 2\eta & , \text{if } \delta < 3 \end{cases}$$

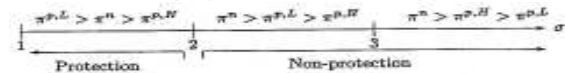
$$\pi^p = \begin{cases} (1+\sigma)\eta^2 & , \text{if } \delta \geq 3 \\ 4\eta^2 & , \text{if } \delta < 3 \end{cases}$$

© 2007 Institute of Information Management

National Chiao Tung University

Software Piracy

- When software users' preferences exhibit network externalities
 - (a) No copy protection yields a higher profit than copy protection if support-oriented consumers place a high value on service offered by the software firm to its legal users, i.e., when $\sigma > 2$
 - (b) Copy protection yields a higher profit than no copy protection when support-



© 2007 Institute of Information Management

National Chiao Tung University

Software Pricing and Market Segmentation

- For the purpose of price discrimination, it may become profitable to increase the unit production cost in order to create a market for low-willingness to pay consumers. As a result, it is commonly observed that the version that is more costly to produce is sold at a lower price than the version that is less costly to produce

© 2007 Institute of Information Management

National Chiao Tung University

Software Pricing and Market Segmentation

- Assume that there are η professional users with a utility function :

$$U^p = \begin{cases} (1+\theta)q - p & , \text{ if buys the software with extra features} \\ q - p & , \text{ if buys the software with no extra features} \\ 0 & , \text{ if does not buy the software} \end{cases}$$
- In addition, there are light users whose utility function :

$$U^l = \begin{cases} q - p & , \text{ if buys the software with or without the extra features} \\ 0 & , \text{ if does not buy the software} \end{cases}$$

© 2007 Institute of Information Management

National Chiao Tung University

Software Pricing and Market Segmentation

Complete version

- The number of software buyers and the corresponding profit levels as functions of these two prices are:

$$q = \begin{cases} \eta & , \text{ if } p = (1+\theta)\eta \\ 2\eta & , \text{ if } p = 2\eta \end{cases}$$

$$\pi = \begin{cases} \eta^2(1+\theta) & , \text{ if } p = (1+\theta)\eta \\ 4\eta^2 & , \text{ if } p = 2\eta \end{cases}$$

© 2007 Institute of Information Management

National Chiao Tung University

Software Pricing and Market Segmentation

Complete version

- When only the complete version is sold, the monopoly software firm will charge a high price, $p = p_H = (1+\theta)2\eta$ if $\theta > 3$, in which case only professional users buy this software; and a low price $p = p_L = 2\eta$, if $\theta \leq 3$, in which case the entire market is served.

© 2007 Institute of Information Management

National Chiao Tung University

Software Pricing and Market Segmentation

Two versions

- Prices

$$(1+\theta)2\eta - p \geq 2\eta - p_r \quad \text{or} \quad p - p_r \leq 2\eta\theta$$
- Profit Level

$$\pi = \eta p + \eta p_r - \phi_r = 2\eta^2(1+\theta) + 2\eta^2 - \phi_r$$

© 2007 Institute of Information Management

National Chiao Tung University

Software Pricing and Market Segmentation

- Comparing the profit generated from selling two versions with complete version
- **If $\phi_\gamma < 2\eta^2\theta$,**
 - **The software firm makes a higher profit by selling two versions of software**
 - **And by selling the version that is more costly to develop at a lower price**



Empirical Findings

- Operation systems and software variety
 - CP/M v.s. MS-DOS
- CD players and variety of CD titles
- Software piracy
 - Through word-of-mouth interactions, pirates may influence the potential users to adopt software, and some of these adopters may eventually purchase the software
 - The percentage of pirated unit adoptions due to the influence of buyers decreased over time stabilizing at around 15% in 1987
- The future of software
 - Compatibility
 - Accessibility
 - Installation and upgrading
 - Flexible payments



Thanks

