KNOWLEDGE TRANSFER IN R&D OUTSOURCING (AND LINUX–VS–WINDOWS)

SALVATORE MODICA*

ABSTRACT. Why did Microsoft not hire all those smart programmers who ended up developing Linux through the internet? Because, we answer, the value of the information about its operating system that Microsoft should have transferred to any of them to render her productive would have been too high compared to her expected individual contribution, so that after writing a contract with Microsoft the typical developer would have run away to sell the acquired knowledge on the market.

On the other hand, knowledge transfer in R&D outsourcing is not always so critical, and for example in the pharmaceutical and chemical industries research contracts are extensively used, usually in the context of a long term relationship between firm and innovator. We analyze this kind of repeated interaction, and find that when the knowledge–transfer problem is not blocking, the firm should transfer to the innovator as much information as it is compatible with the latter's incentive constraints.

JEL Classification System: ...

1. INTRODUCTION

By R&D outsourcing we mean a firm contracting out research about technological advancement of its product. The practice develops during the nineties in response to the need to expand research capabilities in the face of increasing competitive pressure. As both Kimsey–Kurokawa [13] and Holmström–Roberts [12] report, we stress that one often observes strategically critical research being contracted out, as in the case of product design in automobile industry. Of this "something of a trend today toward disintegration, outsourcing, contracting out, and dealing through the market" ([12] p.80) one finds more circumscribed evidence in Thayer [24] and Birch [5] respectively for the chemical and the pharmaceutical industry, the latter reporting that the R&D outsourcing market has grown at an average annual rate of 14.6% between 1997 and 2001. But in other sectors, for example in the case of development of a computer's operating system, the situation is different. In the present paper a unified explanation of this difference will be given.

From a theoretical point of view the only paper dealing with the problem is to our knowledge Aghion–Tirole [1], who study optimal allocation of property rights on innovation in a one–shot interaction between firm and innovator; they conclude that control should be allocated to the innnovator, in terms of the present paper that R&D outsourcing is more efficient than internal product development, if the innovator's effort is 'important enough' (ibidem p.1191). Assuming that this is the case, and also that there are no problems with ex-ante definability of the nature of innovation, we highlight a further potential obstacle to technology outsourcing,

* November 2003, *First Draft.* FINANCIAL SUPPORT: MIUR. AFFILIATION: Dip. Scienze Economiche Finanziarie e Aziendali, Università di Palermo, Viale delle Scienze, 90128 Palermo, Italy. email modica@unipa.it. THANKS: I am grateful to Nicola Persico for inspiring comments and suggestions he has given me in conversations related to the topics of the present paper, and to Enrico Minelli for helping me with its presentation.

which is the following: to render the innovator productive, the firm may have to transmit her some information about its existing technology and internal processes; but this information may be valuable to the innovator independently of her relationship with the firm, possibly so much that she might just walk away with it and default on her contractual obligations with the firm. The remedy to this is to establish a long term relationship, and in practice it often works —but not always. [1]

We then set up and study a multi–period model. The paper has two parts, both stemming from analysis of the above knowledge–transfer problem: the first concerns situations where the severity of the problem impairs the relationship between the firm and potential outside innovators (and there the Linux–vs–Windows issue comes up); the second derives optimal dynamic contracts in cases where the resulting incentive–constrained problem has non–trivial solutions.

First Part: impossibility of outsourcing (section 3). Knowledge management is a problem of increasing practical relevance. Rajan–Zingales [19], who make it one of the main ingredients of their theory of the firm, report for example that Intel was founded by two senior managers of Fairchild Semiconductors who left the latter with an important piece of new technological knowledge acquired thanks to their position; more generally, they quote recent empirical research finding that this phenomenon is rather widespread. Information leaking being a problem within a firm, it can only get worse when it comes to transferring knowledge to third parties. In the extreme it can become a deadlock, when critical bits of information need to be revealed to give the outside innovator the chance to produce valuable results. As it was seen above, this is not the case in the chemical and the pharmaceutical industries. It is, arguably, the case where development of the kernel (core) of a computer operating system is involved: to work on it you have to know it all. [2]

The impossibility part of the paper applies the last observation to interpret the most important novelty in recent history of the computer software industry, namely the challenge to the essential monopoly of the Microsoft Windows operating system posed by the emergence of Linux, a non–proprietary system developed spontaneously through the internet on the basis of an initial creation of a Finnish student called Linus Torvalds. We shall briefly summarize some information on the economic history of Linux in Appendix B; here, being perhaps redundant we would like to stress that the challenge is real. The reader is probably using Microsoft software to read this paper, but currently the crucial penetration is occurring at 'carrier' level (that of large telecommunication corporations); let me bring up two more examples from the business world to illustrate. Hewlett Packard Director of Carrier-Grade Server Christine Martino told *Internetnews* last June that of the customers recently asked about their plans "about 80 to 90 percent [...] are either running Linux or are working on running Linux in the near term." And Motorola, who has already produced a mobile phone powered by Linux, has declared that Linux will serve as a 'key pillar' of its handset software strategy —and has 'put its money where mouth is' by selling its 19 percent stake in Symbian, a leading developer of operating systems for smart phones that use next-generation cellular networks. [3]

---

[1]On the long–term nature of R&D outsourcing relationships in the U.S. and Japan see Birch [5], Kimsey–Kurokawa [13] and Holmström–Roberts [12]; more about this later.

[2]On this point we are in disagreement with Aghion–Tirole [1] who lump together software and biotechnology (overlooking, from our point of view, the knowledge–transfer problem): "When intellectual inputs dominate as for software and biotechnology, research will often be performed by independent units" (ibidem, p.1206).

[3]Web references are respectively `www.internetnews.com/infra/article.php/221613` and `http://news.zdnet.co.uk/hardware/mobile/0,39020360,39117339,00.htm`. *All web sites quoted in the present paper have been visited in October 2003.*

In fact Microsoft itself is the first to be convinced of the seriousness of the threat; see the 'Halloween Documents' at `http://www.opensource.org/halloween`.

But if Linux is becoming the 'best' operating system around, then the question is "How come it has improved so much compared to Windows in the last years?" The easy answer, "Because thousands of developers and an even larger number of testers/debuggers have worked on Linux, while only a few hundreds were working on Windows", only leads to the real economics problem, which is: "Why did Microsoft not hire all those smart programmers who ended up developing Linux, by ad hoc outsourcing contracts?" Certainly not because "It did not know them", for it would not have been difficult to screen the good programmers on the market. The answer we propose is that it did not hire them because of a fatal knowledge–transfer problem.

Second Part: dynamic outsourcing (section 4). The key to overcoming the above impasse and reach a cooperative outcome is, as often in such cases, repetition. On the importance of the long term nature of real outsourcing contracts in Japan see Kimsey–Kurokawa [13]; in business language this becomes 'establishing a preferred–client relationship', see Birch [5] reporting on the success of this strategy in the pharmaceutical industry in the U.S. [4] In the language of the present paper, in this sector the knowledge transfer needed to establish an R&D outsourcing relationship has not been too high compared to the productivity of the innovators.

Focusing attention on the knowledge–transfer problem in a multiperiod setup: given that some knowledge–transfer plan is feasible, and given that the more knowledge is transferred the more the innovator is productive and therefore the bigger is the value to be shared, it is then intuitively obvious that optimally, the firm should transfer as much knowledge as it is compatible with the innovator's incentive constraints. This principle will be seen to indeed hold (just as obviously, when dealing with a dynamic, infinite–dimensional, problem some work is needed to translate intuition into formal statements and proofs). The finding also has some empirical substantiation. Besides Kimsey–Kurokawa [13], also Holmström–Roberts [12], reporting research on automobile industry in the U.S. and Japan, stress the long term nature of interactions but also the 'rich information sharing' (ibidem p.82) between contracting parties.

The sequel of the paper is organized as follows: the formal problem we study is stated in the next section, and the two aforementioned parts of the paper are contained in sections 3 and 4 respectively; the concluding section 5 is about 'lessons from open source' and policy speculations. Proofs are in Appendix A, and some background on the Linux project in Appendix B for ease of reference.

2. Statement of the Problem

There are two actors in the model, a firm and an innovator. The firm knows that the innovator can improve its performance if she has some knowledge of its internal processes. Abstracting altogether from uncertainty, we assume there is an exogenous process of knowledge in the hands of the firm, $\{K_t\}_{t=0,1\ldots}$, which although not formally required, we imagine as being increasing. If the firm transfers knowledge k_t to the innovator, where necessarily $0 \leq k_t \leq K_t$, then net benefit from the latter's work is $V_t(k_t)$; on the other hand the innovator has also the option of just walking away with k_t instead of working for the firm, and if he does this he gets a default value $D_t(k_t)$ —all at time t. At each t, the firm chooses k_t and

[4] The cited article by Birch is 'based on' a longer report, on sale for EUR 1,283.00, which I have not seen. We recall again that besides repetition there are cooperation and integration, on which we do not focus here.

$(\phi_t, 1-\phi_t)$, where ϕ_t and $1-\phi_t$ are its own and the innovator's shares of V_t. The firm's problem we formally analyse is then the following (with discount factor β):

$$\max_{\{k_t,\phi_t\}_{t\geq 0}} \sum\nolimits_{t\geq 0} \beta^t \, \phi_t V_t(k_t)$$

subject to, for all t,

$$0 \leq k_t \leq K_t, \quad \phi_t \leq 1, \tag{P}$$

$$\sum\nolimits_{s\geq t} \beta^{s\text{-}t}(1-\phi_s)V_s(k_s) \geq D_t(k_t)\,, \quad \text{and}$$

$$\sum\nolimits_{s\geq t} \beta^{s\text{-}t} \, \phi_s V_s(k_s) \geq 0.$$

The last two constraints are the innovator's incentive constraint and the firm's participation constraint, which do not need much explanation (the reader may consult Ray [20] for discussion). Finally, we imagine the innovator to be liquidity constrained, and not the firm; this is why it is assumed $\phi_t \leq 1$ but not $\phi_t \geq 0$. V_t is assumed increasing concave, D_t increasing convex, and $V_t(0) = D_t(0) = 0$, for all t. The sequence $\{K_t\}$ is taken to be bounded, so the problem is set up in ℓ_∞ (and its dual; details about this are in appendix).

We are well aware that we are studying a much simplified problem. We have in mind in particular the issues addressed by Aghion–Tirole [1] of contractibility of effort and ex–ante definability of innnovation, which in a dynamic setting become highly relevant hold–up problems (How can the firm walk away if it suspects that the innovator's effort is too low, with strategic knowledge already in the latter's hands? See Kultti–Takalo [14] for a concrete instance of this). Also, equally important is uncertainty in this context, in theory as well as in practice; indeed, uncertainty about the quality of the innovator is one of the main concerns in business practice. [5]

Problem (P) above has the same structure of the one studied by Ray [20]; we have more special strategy sets (forms of agreements), but on the other hand impose no stationarity on the functions involved; and, while Ray identifies a general qualitative property of the solutions to this type of problems, we explicitly solve the one in hand (when the feasible set is non-empty) for a class of cases.

In fact for this this problem the economics of the situations where the problem has an effectively empty feasible set is as interesting as the mathematics of the solutions when the latter exist, so we discuss the two cases separately. In the sequel it will sometimes be more natural to call the two actors principal and agent.

3. No–contract Feasible set: Linux–vs–Windows

The feasible set in problem (P) page 4 is never empty, because it contains all sequences $\{\phi_t, k_t\}$ with $k_t = 0$ all t (and $\phi_t \in [0,1]$); if it contains no other points, the only feasible contract between firm and innovator is the null contract. The conditions under which this is the case are easily spelled out. The extreme case is with $V_t \equiv 0 \; \forall t > 0$ and $V_0 < D_0$, and the general case is then clear: the innovator's productivity falls rapidly with time, and the first–shot outcome is not as valuable as the knowledge needed to produce it. This will be the case for example if the innovator's productivity is low for all but near–full knowledge transfer, and for such transfers on the other hand the default value is very high. [6]

---

[5]In the pharmaceutical and biopharmaceutical industry for example, the difficulty starts with choosing among a large number of 'Contract Research Organizations' (what we call innovators) —cfr. Thomas Kupiec, "Analytical Outsourcing: assessing outsource/in–house options", *Contract Pharma* March 2003, online at `http://www.contractpharma.com/March032.htm`.

[6]There still is the alternative of in–house development, 'integration' in the terminology of Aghion–Tirole [1], when outsourcing is unfeasible; but in the Linux–Windows context on which

Going back to the question raised in the introduction of interpreting the recent Linux–vs–Windows history (and in terms of the next section's 'past' and 'present' we are here interpreting *past* history), at this point there is little to say that the reader has not already anticipated: it is our contention that the above no–contract scenario fits well the story under discussion. Indeed, to develop an operating system a programmer cannot do without knowing it deeply. And moreover, it is often by inspecting its various aspects that she finds the one whose improvement best fits her capabilities —as in science. Finally, and the parallel to science is again inevitable, often a programmer's best shot is his first; in the above terms V_t decreases sharply with time.

The other part of the argument is that the magnitude of the default value D_0 may well be high compared to V_0 for near–full knowledge transfer, and the rationale for this is that, as the Fairchild–Intel story recalls, the value of *critical* knowledge may be in the order of the value of the firm itself. If this is the case, in one–developer firms the order of magnitude of $D_0(K_0)$ is the discounted sum of all future $V_t(K_t)$'s; when many developers are employed, it is a lot higher. In particular, it seems reasonable to think that the value of the source code of the Windows operating system is uncomparably higher than the potential value of a developer's contribution. [7]

So to repeat what anticipated in the introduction, our interpretation of the technical explosion of the Linux operating system with little or no comparable reaction on the part of the incumbent Microsoft monopolist is that the latter, owing to the above described knowledge–transfer problem, was forced to hang on to its few hundreds of programmers (and debuggers), and was therefore totally unfit to compete with the army of open–source developers and testers who contributed, with no problems of that sort, to make Linux the powerful operating system which it is today. We stress that there was no knowledge–transfer problem in the Linux side; the reason is simply that knowledge there is common property; we expand on this latter point (and some related ones) in Appendix B.

4. Non–trivial Feasible Set: Optimal Outsourcing

We now assume that the zero knowledge–transfer path is dominated by other feasible alternatives, and begin to analyse the problem by writing the Lagrangean and imposing stationarity and complementary slackness conditions. Justification of this procedure is in the appendix.

To anticipate, there are no surprises in the solutions. Inspection of problem (P) page 4 quickly reveals that principal and agent have here a common interest —that V be as high as possible. Thus intuitively the solution should call for as large a transfer of k as it is compatible with the agent's incentive constraint. This is what formal analysis confirms.

The Lagrangean of problem (P) without the non-negativity constraints on k_t is

$$\mathcal{L} = \sum\nolimits_{t\geq 0} \beta^t \Big[\phi_t V_t(k_t) + \xi_t(K_t - k_t) + \zeta_t(1-\phi_t)\Big]$$
$$+ \sum\nolimits_{s\geq 0} \lambda_s \Big[\sum\nolimits_{t\geq s} \beta^{t\text{-}s}(1-\phi_t)V_t(k_t) - D_s(k_s)\Big] + \sum\nolimits_{s\geq 0} \mu_s \Big[\sum\nolimits_{t\geq s} \beta^{t\text{-}s}\, \phi_t V_t(k_t)\Big].$$

we shall focus attention the same conditions which lead to the null–contract feasible set also make integration unprofitable, for the latter would mean employing thousands of programmers, with long term costs by far exceeding expected benefits.

[7] Of course the type of knowledge the firm has to transfer to the innovator does not *need* to be 'critical'; this is indeed what the successful stories of technology outsourcing demonstrate. It is worth noting that in such cases the firm often has *multiple* contractors at the same time; for the biotechnology industry this is reported in research quoted by Holmström–Roberts [12] (p.85–86).

Letting

$$\nu_t = \lambda_0\beta^t + \lambda_1\beta^{t-1} + \cdots + \lambda_t, \quad \rho_t = \mu_0\beta^t + \mu_1\beta^{t-1} + \cdots + \mu_t$$

this becomes

$$\mathcal{L} = \sum\nolimits_{t\geq 0} \beta^t \Big[\xi_t(K_t - k_t) + \zeta_t(1-\phi_t) - \beta^{-t}\lambda_t D_t(k_t)\Big] + \sum\nolimits_{t\geq 0} V_t(k_t)\Big[(\beta^t + \rho_t)\phi_t + \nu_t(1-\phi_t)\Big] .$$

Thus complementary slackness and FOC are

$$\xi_t(K_t - k_t) = 0, \quad \zeta_t(1-\phi_t) = 0 \ \ \forall t,$$
$$\lambda_s\Big[\sum\nolimits_{t\geq s} \beta^{t\text{-}s}(1-\phi_t)V_t(k_t) - D_s(k_s)\Big] = 0, \quad \mu_s \sum\nolimits_{t\geq s} \beta^{t\text{-}s}\,\phi_t V_t(k_t) = 0 \ \ \forall s,$$
$$\big[(\beta^t + \rho_t)\phi_t + \nu_t(1-\phi_t)\big]\, V_t'(k_t) - \lambda_t D_t'(k_t) = \beta^t\xi_t\,,$$
$$\big[\beta^t + \rho_t - \nu_t\big]\, V_t(k_t) = \beta^t\zeta_t \ \ \forall t\,.$$

So the FOC with respect to k_t is

$$\begin{aligned}\big[(\beta^t + \rho_t)\phi_t + \nu_t(1-\phi_t)\big]\, V_t'(k_t) - \lambda_t D_t'(k_t) &\geq 0, \\ &= 0 \text{ if } k_t < K_t\,.\end{aligned} \tag{1}$$

And given $V_t(k_t) > 0\ \forall k_t > 0$, the one with respect to ϕ_t is $\beta^t + \rho_t - \nu_t \geq 0$, equal if $\phi_t < 1$; this is more conveniently rewritten as

$$\begin{aligned}(\lambda_0 - \mu_0) + (\lambda_1 - \mu_1)\beta^{-1} + \cdots + (\lambda_t - \mu_t)\beta^{-t} &\leq 1, \\ &= 1 \text{ if } \phi_t < 1\,.\end{aligned} \tag{2}$$

Now observe that $\phi_t = 1\ \forall t$ is not feasible, for it violates the agent's incentive constraint (we are dealing with solutions with non-zero $\{k_t\}$, for which the V_t and D_t functions are non-zero). Let t_0 be the first t such that $\phi_t < 1$. Then from (2), after t_0 the first s such that $\lambda_s \neq \mu_s$ must have $\lambda_s < \mu_s$. This implies that inequality in (2) strict at s, whence $\phi_s = 1$; it also implies $\mu_s > 0$, which by complementary slackness gives $\sum_{t\geq s} \beta^{t\text{-}s}\,\phi_t V_t(k_t) = 0$; and the latter, with $\phi_s = 1$, then implies that the principal participation constraint is violated at $s+1$. Conclusion: after t_0 one has $\lambda_t = \mu_t\ \forall t$. Therefore, for all $t > t_0$: either (i) $\lambda_t = \mu_t = 0$; or (ii) $\lambda_t = \mu_t > 0$.

In case (i), $\nu_t - \rho_t = \beta^t$ (for all $t > t_0$): indeed at t_0, since $\phi_{t_0} < 1$, equation (2) says $\nu_{t_0} - \rho_{t_0} = \beta^t$, so $\nu_{t_0+1} - \rho_{t_0+1} = \beta(\nu_{t_0} - \rho_{t_0}) + (\lambda_{t_0+1} - \mu_{t_0+1}) = \beta^{t_0+1}$, etc. Thus (1) becomes $(\beta^t + \rho_t)V_t'(k_t) \geq 0$, equal if $k_t < K_t$. But both factors on the left side of the inequality are positive, so $k_t < K_t$ cannot be: in this case $k_t = K_t\ \forall t > t_0$.

In case (ii), from the complementary slackness conditions on λ and μ we get, for each $s > t_0$,

$$\sum\nolimits_{t\geq s} \beta^{t\text{-}s}\,\phi_t V_t(k_t) = 0 \ \text{ and } \ \sum\nolimits_{t\geq s} \beta^{t\text{-}s}\, V_t(k_t) = D_s(k_s)\,.$$

The first set of equalities easily imply that $\phi_t = 0\ \forall t > t_0$. So all production value goes to the agent after t_0, and $\{k_t\}_{t>t_0}$ is chosen so that this value just covers default value D_t at each t (recalling that V_t are concave and D_t convex, this means that he chosen k's could not be higher without violating the agent's incentive constraint). In this case the structure of the solution is the same as that found by Ray in [20] in the stationary case.

In both cases, the picture for $t < t_0$ is that since $\phi_t = 1$, the principal's individual rationality constraints are met with strict inequality, whence $\mu_t = 0$; so (1) becomes

$$\beta^t V_t'(k_t) \geq \lambda_t D_t'(k_t), \qquad = \quad \text{if } k_t < K_t.$$

We next explicitly describe the solution in three special cases, differing in the amount of knowledge which can be transferred in equilibrium. Letting

$$\mathcal{V}_t = \sum\nolimits_{s \geq t} \beta^{s\text{-}t}\, V_s(K_s)\,,$$

we consider the situations where one of the following conditions holds for all t:

$$D_t(K_t) = \mathcal{V}_t; \quad D_t(K_t) > \mathcal{V}_t; \quad D_t(K_t) < \mathcal{V}_t.$$

In the last case, intuitively the solution should have $k_t = K_t$ all t, and we shall confirm formally that this is so, except possibly for a finite number of initial periods. In the second case such full knowledge transfer is clearly not feasible: the feasible set can be 'thin', so the first step is to impose conditions which guarantee that it is large enough for the problem to be interesting; here stationarity seems to be the single most natural assumtption to make. The first case is obviously a measure–zero set, but it is instructive for the solutions of the others, and with it we start.

Case $D_t(K_t) = \mathcal{V}_t\ \forall t$. We shall see that this case falls under heading (ii) above. We make here *two assumptions*. First, in keeping with the spirit of the present section that no–knowledge–transfer is dominated, we assume that $V_0'(0) > D_0'(0)$; since $D_0(0) = V_0(0) = 0$ and $D_0(K_0) = V_0(K_0) + \beta\mathcal{V}_1 > V_0(K_0)$, we then have

$$0 < \text{argmax}_{k_0}\, V_0 - D_0 < K_0. \tag{3}$$

The second is a more technical assumption, quite natural in the ℓ_∞ setup:

$$\textit{The sequence } (\mathcal{V}_t)_{t\geq 0} \textit{ is bounded.} \tag{4}$$

In the next proposition optimal policy is characterized. The idea is that the principal wants to push knowledge transfer k_t up to K_t as soon as she can and leaving all value to the agent from then on (feasibility and $k_t = K_t$ implies $\phi_t = 0$), conditional on being able to appropriate the value coming out of the initial phase. The optimal policy would be to do this in period 1 if the agent were not liquidity constrained (that is if there were no constraint $\phi_0 \leq 1$); with this constraint the initial phase lasts usually longer.

To get some intuition we may start by observing that the agent's incentive constraint at $t = 0$ is just

$$\sum\nolimits_{t\geq 0} \beta^t\, \phi_t V_t(k_t) \leq \sum\nolimits_{t\geq 0} \beta^t\, V_t(k_t) - D_0(k_0)\,; \tag{5}$$

so if the right member is maximized subject to the other constraints and to being equal to the left member (which is the principal's payoff), the solution to (P) is found. To see how the optimization process goes observe that the highest value the right side of (5) can take is $[\max_{k_0} V_0 - D_0] + \beta\mathcal{V}_1$; and that setting $k_t = K_t\ \forall t \geq 1$ forces $\phi_t = 0\ \forall t \geq 1$ by the assumption $D_t(K_t) = \mathcal{V}_t$; the latter also implies that the sequence $\phi_t = 1, k_t = K_t\ \forall t \geq 1$ satisfies all incentive constraints for such t's. Therefore if there a $\phi_0 \leq 1$ such that the incentive constraint at $t = 0$ holds with equality, that is such that

$$\phi_0 V_0(k_0) = V_0(k_0) - D_0(k_0) + \beta\mathcal{V}_1 \quad \text{with} \quad k_0 = \text{argmax}(V_0 - D_0),$$

then (5) holds with equality and the problem is solved: $k_0 = \text{argmax}(V_0 - D_0)$, $k_t = K_t\ \forall t \geq 1$, ϕ_0 defined by the last equation displayed, and $\phi_t = 0\ \forall t \geq 1$. In this case the principal obtains all of her payoff in the first period. Problem is that a $\phi_0 \leq 1$ as required above may not exist, that is, it may be that

$$D_0(k_0) \geq \beta \mathcal{V}_1, \quad k_0 = \text{argmax}(V_0 - D_0) \tag{6}$$

fails. In the latter case the next step is to try and maximize the right member of (5) with respect to k_0 and k_1, leaving $k_t = K_t\ \forall t \geq 2$, while respecting the agent's incentive constraints and subject to finding ϕ_0 and ϕ_1 ($\phi_t = 0\ \forall t \geq 2$) such that at the maximizing values of k_0, k_1 relation (5) hold with equality. In this problem, optimal $\phi_0 = 1$; so if the optimal *unconstrained* value of ϕ_1 is ≤ 1 then again we have found the solution of (P), and the principal gets all her payoff in the first two periods. We shall show in appendix that the process stops in a finite number of stages, and this leads to the proposition which follows.

As to the statement below, observe that given $D_t(K_t) = \mathcal{V}_t$, problem (P) reduces to the one appearing there when after t_0 one imposes $\phi_t = 0$ and $k_t = K_t$ (and neglects $\phi_{t_0} \leq 1$). Again, proof is in appendix.

Proposition 1. *In the case $D_t(K_t) = \mathcal{V}_t\ \forall t$, assuming $V_0'(0) > D_0'(0)$ and that the sequence $(\mathcal{V}_t)_{t\geq 0}$ is bounded, there exists a first time $t_0 < \infty$ such that the problem*

$$\max_{\{k_t,\phi_t\}_{t=0,\dots,t_0}} \sum_{t=0}^{t_0} \beta^t V_t(k_t) + \beta^{t_0+1}\mathcal{V}_{t_0+1} - D_0$$

subject to

$$k_t \leq K_t,\ t = 0,\dots,t_0 \quad \phi_t \leq 1,\ t = 0,\dots,t_0 - 1$$

$$\sum_{s=t}^{t_0} \beta^{s-t}\phi_s V_s = \sum_{s=t}^{t_0} \beta^{s-t} V_s + \beta^{t_0+1-t}\mathcal{V}_{t_0+1} - D_t,\ t = 0,\dots,t_0$$

has optimal $\phi_{t_0} \leq 1$ (from the last of the above constraints, this ϕ_{t_0} is defined by the equation $\phi_{t_0} V_{t_0}(k_{t_0}) = V_{t_0}(k_{t_0}) + \beta\mathcal{V}_{t_0+1} - D_{t_0}(k_{t_0})$). The problem also has optimal $\phi_t = 1$ for all $t < t_0$.

Optimal policy for problem (P) is given by the solution to the above problem, followed by $\phi_t = 0$ and $k_t = K_t$ for all $t > t_0$.

Remark. It is shown in the proof that for all $t < t_0$ the optimal policy has $D_t(k_t) = \beta D_{t+1}(k_{t+1})$. Thus a sufficient condition for the optimal amount of knowledge transfer to be *increasing* is that $\beta D_{t+1} < D_t$ for all t, i.e. that the default value does not grow too fast.

Case $D_t(K_t) > \mathcal{V}_t\ \forall t$. This case also falls under heading (ii) of page 6. This is the case where vaule of the default option is relatively high, and so it is the closest to the no–contract feasible set. To guarantee that a non–trivial set of contracts is feasible, as we said before we have found that imposing stationarity is the single more natural assumption. We then *assume here* that:

$$V_t = V, \quad D_t = D, \quad K_t = K, \quad \forall t \geq 0. \tag{7}$$

Thus the inequality $D_t(K_t) > \mathcal{V}_t\ \forall t$ becomes

$$D(K) > (1-\beta)^{-1}V(K).$$

Given this, $V(0) = D(0)$ and concavity of $V - (1-\beta)D$, if the derivative at zero of the latter function is non–positive the only stationary sustainable level of knowledge transfer is zero even with $\phi = 0$, because it would be $D(k) > (1-\beta)^{-1}V(k)\ \forall k > 0$. We then *also assume* that $V'(0) > (1-\beta)D'(0)$ (this is weaker than the corresponding assumption in the previous case). Hence there exist a largest level of stationary sustainable knowledge transfer $0 < k^* < K$, defined by

$$D(k^*) = (1-\beta)^{-1}V(k^*). \tag{8}$$

As in the previous case the idea is that the principal wants to push up k_t as soon as she can; but now not up to K, which is unfeasible, but up to k^*.

Proposition 2. *In the case $D_t(K_t) > \mathcal{V}_t$ $\forall t$, assuming stationarity* (7) *and that $V'(0) > (1-\beta)D'(0)$, all of proposition 1 can be restated, with the following two modifications: $\mathcal{V}_{t_0+1}$ in the problem there described replaced by $(1-\beta)^{-1}V(k^*)$, and the final statement "$k_t = K_t$ for all $t > t_0$" changed to "$k_t = k^*$ for all $t > t_0$".*

Moreover, in the present case optimal knowledge transfer is increasing (for the sufficient condition stated in the remark following proposition 1 holds here).

Case $D_t(K_t) < \mathcal{V}_t$ $\forall t$. This case will fall under heading (i) of page 6. As we already observed one would guess that full knowldege transfer is optimal; this is only partially true, because as we shall see it may happen that full transfer does not begin at time zero. To ease exposition we shall again make two simplifying assumptions. The first is in the spirit of stationarity:

$$\mathcal{V}_t = \mathcal{V} \ \forall t \geq 0. \tag{9}$$

This is equivalent to assuming $V_t(K_t) = V_0(K_0)$ $\forall t \geq 0$ ((9) is obviously implied by the latter; but given (9) one has $\mathcal{V}_t = V_t(K_t) + \beta\mathcal{V}_{t+1} = V_t(K_t) + \beta\mathcal{V}_t$ whence $V_t(K_t) = (1-\beta)\mathcal{V}_t = (1-\beta)\mathcal{V}$ $\forall t$). Next, although default value is 'small' here, we still find it natural to imagine that the gap between $\mathcal{V}$ and $D_t(K_t)$ would shrink with time. We assume that this occurs at a constant rate, in the sense that for some $\gamma \in (0,1)$ one has

$$\mathcal{V}_t - D_t(K_t) = \gamma\big[\mathcal{V}_{t-1} - D_{t-1}(K_{t-1})\big], \quad t > 0, \tag{10}$$

which given (9) is obviously equivalent to $D_t(K_t) = (1-\gamma)\mathcal{V} + \gamma D_{t-1}(K_{t-1})$.

Now define

$$\hat{\phi}_t = \frac{1-\beta\gamma}{1-\beta}\,\frac{\mathcal{V} - D_t(K_t)}{\mathcal{V}}\,.$$

Since $\mathcal{V} - D_t(K_t) = \gamma^t\big[\mathcal{V} - D_0(K_0)\big]$, clearly $\hat{\phi}_t = \gamma^t\hat{\phi}_0$. Call $t_0 \geq 0$ the first t such that $\hat{\phi}_t \leq 1$. Optimal policy is then as follows (proof in appendix):

Proposition 3. *In the case $D_t(K_t) < \mathcal{V}_t$ $\forall t$, under assumptions* (9) *and* (10)*, for the $t_0 \geq 0$ just defined, optimal policy has $k_t = K_t$ and $\phi_t = \hat{\phi}_t$ $\forall t > t_0$. Optimal k_t and ϕ_t for $t \leq t_0$ are specified in the appendix.*

For example, if $\hat{\phi}_{t_0} = 1$ the policy for this initial phase is $k_t = K_t,\ \phi_t = 1\ \forall t \leq t_0$; otherwise the latter is usually not feasible (again details in appendix).

Notice that in the present case the principal's continuation payoff, although decreasing to zero, remains positive forever.

5. Conclusions

As anticipated in the introduction we have no conclusions, and this section only contains some comments on what we may abstract from the open source experience, that is, on the conditions under which an open source model might be applied outside software production. [8]

Core and Trusted Authority. The fact that it is essential that there be a 'substantive initial core' which has the potential to become of widespread use is well recognized (cfr. e.g. Benkler [3], Weber [29]). And from the organizational point of view I would like to stress the importance of a 'central authority' (like Linus in person at the beginning of the Linux project and the OSDL these days).

Product Cycle and Quality Circles. To peer–develop an initial product each contributor must obvously have a higher payoff from revealing than from concealing

[8] For ease of reference some background on open source is collected in Appendix B.

his work (and using it only for himself). Assuming the possibility to emarginate those who conceal from sharing in subsequent improvements, the requirement is that there be an equilibrium in which each reveals her contribution to the community and benefits from the others' ones (the 'contributing a brick to have a complete house in return', cfr. p. 17). The conditions for this seem to be most favourable: (i) for complex products, where improvements occur in all directions/parts; [9] and (ii) in the initial phase of development, where due to decreasing returns to research the value of individuals' contributions is highest.

User Value and Granularity. On the motivational side one surely cannot rely on social excitement (cfr. Appendix B), but the same must be said of the often quoted signalling motive (see e.g. Lerner–Tirole [16]): as reported in the Appendix, after close inspection it was found rather weak even in the software case. What remains is the user–value, which as we learn from the work of von Hippel should not be undervalued. It is to be stressed that one should have in mind here the firm–user more than the consumer–user: a firm innovating a process machine which it uses in a major product line may make substantial profits from the innovation; or, think of brakes improvements on the part of a producer of racing cars or aircrafts. Thus, 'large team work' is not necessarily an insurmountable barrier. All the more so given that in a peer production process cooperation has no complicated property–related drawbacks; see, most notably, the current experience of the OSDL (see again Appendix B) where all major competitors in the telecommunication market are involved.

Intermediate Products. The discussion so far points to a specific class of goods: the intermediate products. A moment's thought suggests the qualification that they should not be crucial for gaining competitive edge; for example, brakes but not components/solutions affecting fuel consumption for family cars in Europe (where fuel is highly taxed and fuel consumption is the first thing consumers watch and car makers advertise).

Timing. The Linux kernel has develped so fastly because 'the world' was just ready to take up Linus' work and improve upon it. Quite likely, if Linus had written his 0.01 version in 1975 instead of 1991, history would have been very different. In fact we have an example of something like this happening: John von Neumann's insights on computers' architecture date 1945, but it took about ten years before they influenced industrial production (which they did pervasively when time was ripe; cfr. Mowery–Rosenberg [18]). The point here is obviuos but it may be crucial: for peer production, peers must be there and ready.

Policy and Initial Core. This is not a paper about intellectual property and desirability of patents, and we will not raise the point in the last paragraph. See Bessen–Maskin [4] and Scotchmer [23] for 'problems' with patents in the presence of cumulative innovation. Remaining focused on peer production, we remark that the weak spot in such processes which the preceeding observations point to is the existence of the 'substantive initial core' —there are not many Torvalds or von Neumann around.

Given an initial core, under the conditions just spelled out product development is possibly faster in a patent–free, open source environment rather than in a proprietary system; but as we all know absence of patents may deter production of primary innovations/initial cores. Thus the trade-off for growth which seems to emerge for patent policy is between having more primary innovations with slower improvement against having fewer innovations with more extensive development.

[9] I speak of directions here because for such products I visualize, in place of the traditional 'quality ladder' (cfr. Scotchmer), an image of quality (lager and lager) 'circles'.

The additional policy dimension may be the public funding of open sourced primary innovations whenever this seems propitious; the above considerations are intended to contribute to identify conditions under which it may be so.

Appendix A: Mathematical Arguments

Justifying the Lagrangean. There are two steps involved in writing the Lagrangean the way we have done (i.e. the 'usual' way) and imposing its stationarity in this context. The first concerns existence of multipliers in the dual of ℓ_∞; the second regards conditions ensuring that those multipliers are in fact in ℓ_1 (a subset of the above dual), i.e. expressible as a sequence of real numbers. For both we shall merely invoke existing theorems.

Existence of multipliers in the positive cone of ℓ_∞^* such that at the optimal solution the lagrangean is stationary follows for instance from theorem 1.10 of chapter 3 (p.190) of Barbu–Precupanu [2]. The regularity condition (ib. p.191) in our case amounts to the requirement that the inequality

$$\sum\nolimits_{t \geq s} \beta^{t\text{-}s}(1-\phi_t)V_t' > D_s'$$

hold for s sufficiently large along the optimal sequence. Conditions ensuring this are easy to write in all cases considered in the paper (a simple, but unappealing one is that β be close enough to one).

As to the ℓ_1 problem, we can apply corollary 5.6 of Rustichini [22] directly; validity of its hypotheses in our case is immediate to check.

Proof of Proposition 1. We resume the argument where it was interrupted, after (6). As we were saying, if that relation fails one turns to the two–period problem, which is the following:

$$\max_{(\phi_0,\phi_1),(k_0,k_1)} V_0 + \beta V_1 + \beta^2 \mathcal{V}_2 - D_0$$

subject to

$$\phi_0 \leq 1, \quad k_t \leq K_t,\ t = 0, 1 \qquad (P_2)$$

$$\phi_0 V_0 + \beta\phi_1 V_1 = V_0 + \beta(V_1 + \beta\mathcal{V}_2) - D_0 \qquad (a)$$

$$\phi_1 V_1 = V_1 + \beta\mathcal{V}_2 - D_1. \qquad (b)$$

Substituting the last constraint the problem can be written as

$$\max_{(k_0,k_1)} V_0 + \beta\left[V_1 + \beta\mathcal{V}_2\right] - D_0$$

$$\text{subject to} \quad D_0 \geq \beta D_1, \quad k_t \leq K_t,\ t + 0, 1. \qquad (P_2')$$

Suppose that the solution to this problem has $D_1 \geq \beta\mathcal{V}_2$; then the value of ϕ_1 defined by (b) satisfies $\phi_1 \leq 1$, so that (ϕ_0, ϕ_1) defined by (a) and (b) together with the solution (k_0, k_1) of (P_2') solve (P): to wit, the solution of the latter is $\phi_0 = 1$ (which follows from the fact that the constraint $D_0 \geq \beta D_1$ in (P_2') is binding, as will be verified shortly), ϕ_1 defined by (b), (k_0, k_1) solving (P_2'), and $\phi_t = 0$, $k_t = K_t$ $\forall t \geq 2$. In this case the principal gets her payoff in the first two periods, and from then on only the agent's payoff is positive.

If the solution to (P_2') has $D_1 < \beta\mathcal{V}_2$, then we pass to the obvious next step, which is the three–period try. We shall show that this process ends in a finite number of steps, but before moving on we must check that the constraint $D_0 \geq \beta D_1$ is binding in (P_2'). The lagrangean is

$$V_0 - D_0 + \beta\left[V_1 + \beta\mathcal{V}_2\right] + \lambda(D_0 - \beta D_1) + \mu_0(K_0 - k_0) + \beta\mu_1(K_1 - k_1);$$

so FOC and complementary slackness give

$$V_0' - (1-\lambda)D_0' \geq 0, \quad = 0 \text{ if } k_0 < K_0, \tag{$F_2\, i$}$$

$$V_1' - \lambda D_1' \geq 0, \quad = 0 \text{ if } k_1 < K_1. \tag{$F_2\, ii$}$$

If $k_1 < K_1$ then from $(F_2\, ii)$ $\lambda = V_1'/D_1' > 0$, so the constraint in question binds. Thus it may be non–binding only if $k_1 = K_1$; with this and $\lambda = 0$, $(F_2\, i)$ reads $V_0' - D_0' \geq 0$, $= 0$ if $k_0 < K_0$. But since the max of the concave function $V_0 - D_0$ is interior by (3) one has $V_0' - D_0' < 0$ at K_0, so it cannot be $k_0 = K_0$; so it should be $k_0 < K_0$; but then $V_0' - D_0' = 0$, i.e. $k_0 = \mathrm{argmax}(V_0 - D_0)$; on the other hand, for this pair (k_0, K_1) we have by failure of (6) $D_0 < \beta\mathcal{V}_1 = \beta D_1$, i.e. the pair is not feasible for the problem in hand. We conclude that the constraint is binding, and so optimal $\phi_0 = 1$ in (P_2). We observe for future reference that it has also been shown that $\lambda > 0$; this implies, via $(F_2\, i)$, that $k_0 > \mathrm{argmax}(V_0 - D_0) > 0$.

To see what is involved in showing that the the process ends in a finite number of steps let us look again at the inequality $D_1 \geq \beta\mathcal{V}_2$ in (P_2'); since we have just found $D_0 = \beta D_1$ at the optimum, this is

$$D_0(k_0) \geq \beta^2\mathcal{V}_2, \quad k_0 \text{ solving } (P_2'). \tag{11}$$

Comparing this with (6) we guess that the s-period try will be the successful one if the inequality $D_0(k_0) \geq \beta^s\mathcal{V}_s$ holds for k_0 optimal solution of the relevant problem. Since it will be shown that this k_0 will always be not smaller than argmax $(V_0 - D_0)$, by the boundedness assumption (4) the inequality will be satisfied for s large enough.

We turn to the $(s+1)$–period problem, in the variables $k_0, \dots, k_s$. The hypothesis is that for the sequence $k_0, \dots, k_{s-1}$ solving the s–period problem, one has $D_{s-1} < \beta\mathcal{V}_s$; and that similarly for $k_0, \dots, k_{s-2}$ solving the $(s-1)$–period problem one has $D_{s-2} < \beta\mathcal{V}_{s-1}$; and so on down to the one–period problem. In words, the induction hypothesis is that problem (P) cannot be solved by the principal appropriating all of her payoff in less than $s+1$ periods.

We consider the $(s+1)$–period analogue of problem (P_2) and arrive at the $(s+1)$–period version of problem (P_2'), which is, omitting the constant term $\beta^{s+1}\mathcal{V}_{s+1}$,

$$\max_{(k_0,\dots,k_s)} \sum_{t=0}^{s} \beta^t V_t - D_0$$

subject to (P_s')

$$k_t \leq K_t, \ t = 0, \dots, s$$

$$D_t \geq \beta D_{t+1}, \ t = 0, \dots, s-1.$$

Again our aim is to show that the constraints on D are binding. For then the question whether $\phi_s \leq 1$, i.e. $D_s \geq \beta\mathcal{V}_{s+1}$, becomes $D_0 \geq \beta^{s+1}\mathcal{V}_{s+1}$. The lagrangean for (P_s') is

$$V_0 - D_0 + \lambda_0(D_0 - \beta D_1) + \mu_0(K_0 - k_0) + \beta\left[V_1 + \lambda_1(D_1 - \beta D_2) + \mu_1(K_1 - k_1)\right] + \dots$$
$$+ \beta^{s-1}\left[V_{s-1} + \lambda_{s-1}(D_{s-1} - \beta D_s) + \mu_{s-1}(K_{s-1} - k_{s-1})\right] + \beta^s\left[V_s + \mu_s(K_s - k_s)\right].$$

Stationarity and complementary slackness give

$$V_0' - (1-\lambda_0)D_0' \geq 0, \quad = 0 \text{ if } k_0 < K_0$$
$$V_1' - (\lambda_0 - \lambda_1)D_1' \geq 0, \quad = 0 \text{ if } k_1 < K_1$$
$$\dots\dots\dots$$
$$V_{s-1}' - (\lambda_{s-2} - \lambda_{s-1})D_{s-1}' \geq 0, \quad = 0 \text{ if } k_{s-1} < K_{s-1}$$
$$V_s' - \lambda_{s-1}D_s' \geq 0, \quad = 0 \text{ if } k_s < K_s.$$

As in the two–period case, from the last condition displayed we deduce that for $D_{s-1} - \beta D_s$ not to be binding it must be $k_s = K_s$, and $\lambda_{s-1} = 0$. But then the rest of the conditions are exactly the same as those of the s–period problem, in which case the solution would be the same as that of the s–period problem followed by $k_s = K_s$; but then the hypothesis implies that $D_{s-1} < \beta \mathcal{V}_s = \beta D_s$, contradicting feasibility; so $D_{s-1} = \beta D_s$.

Next $D_{s-2} - \beta D_{s-1}$. If $k_{s-1} < K_{s-1}$ then as before $\lambda_{s-2} - \lambda_{s-1} = V'_{s-1}/D_{s-1} > 0$ which would imply that the constraint is binding. If on the other hand $k_{s-1} = K_{s-1}$ and $\lambda_{s-2} = 0$, we are back to the $(s-1)$–period problem, which with $k_{s-1} = K_{s-1}$ has $D_{s-2} < \beta D_{s-1}$, contradicting feasibility again. So $D_{s-2} - \beta D_{s-1}$ is binding, and continuing this way we conclude that all the D constraints are binding. It has also been shown, incidentally, that always $\lambda_0 > 0$.

Now, as anticipated, given $D_0 = \beta D_1 = \cdots = \beta^s D_s$ (and $\phi_0 = \cdots = \phi_{s-1} = 1$ in problem (P_s)), the question whether $\phi_s \leq 1$, i.e. $D_s \geq \beta \mathcal{V}_{s+1}$, becomes

$$D_0(k_0) \geq \beta^{s+1} \mathcal{V}_{s+1}, \tag{12}$$

k_0 being part of the solution to the s–period problem. And this holds for s sufficiently large. Indeed, in any s–period problem either $k_0 = K_0$, or from complementary slackness $V'_0 - D'_0 = -\lambda_0 D'_0 < 0$, last inequality from $\lambda_0 > 0$; thus in all problems the optimal $k_0 > \operatorname{argmax}(V_0 - D_0)$, whence the left member of (12) is bounded away from zero; on the other hand, by assumption (4) the right member tends to zero as s diverges. This concludes the argument.

Proof of Proposition 2. As we did in the previous case we rewrite the agent's incentive constraints, the first one binding:

$$\sum\nolimits_{t \geq 0} \beta^t \, \phi_t V(k_t) = [V(k_0) - D(k_0)] + \sum\nolimits_{t \geq 1} \beta^t V(k_t) \tag{13}$$

$$\sum\nolimits_{t \geq s} \beta^{t-s} \, \phi_t V(k_t) \leq \sum\nolimits_{t \geq s} \beta^{t-s} \, V(k_t) - D(k_s), \; s \geq 1\,. \tag{14}$$

Forget as before the constraint $\phi_0 \leq 1$. In the previous case it was then immediate that the max the right member of (13) was $[\max_{k_0} V_0 - D_0] + \beta \mathcal{V}_1$, and that this choice of $\{k_t\}_{t \geq 0}$ satisfied (with equality) the other incentive constraints. In the present case the situation is slightly more complex: the choice $k_t = K \; \forall t \geq 1$ is unfeasible, and then maximization of $\sum_{t \geq 1} \beta^t V(k_t)$ with respect to $\{k_t, \phi_t\}_{t \geq 1}$ subject to the constraints (14) is non–trivial. We shall now show that it is solved by $k_t = k^*$, $\phi_t = 0 \; \forall t \geq 1$. Thus if for this choice (with $k_0 = \operatorname{argmax}(V - D)$) the ϕ_0 defined by $\phi_0 V(k_0) = V(k_0) - D(k_0) + \beta(1-\beta)^{-1} V(k^*)$ happens to be ≤ 1, problem (P) is solved.

If not, as before the principal has to try and appropriate his payoff in two periods. In this case again the difference compared to the case $D_t(K_t) = \mathcal{V}_t \; \forall t$ is that we have a non–trivial maximization, of $\sum_{t \geq 2} \beta^t V(k_t)$ under the constraints in (14) for $s \geq 2$; but again it is be proved by the same argument as the one we are about to give that the solution to this maximum problem is $k_t = k^*, \phi_t = 0 \; \forall t \geq 2$. Thus at stage two we are again in a position analogous to that of case $D_t(K_t) = \mathcal{V}_t \; \forall t$, with $\mathcal{V}_2$ replaced by $(1-\beta)^{-1} V(k^*)$ in problem (P_2) of page 11. At this point the argument parallels the previous one: optimal $\phi_0 = 1$, and if the ϕ_1 defined by the equation $\phi_1 V(k_1) = V(k_1) - D(k_1) + \beta(1-\beta)^{-1} V(k^*)$, with k_1 part of the solution of the modified (P_2), is ≤ 1, then problem (P) is solved. Otherwise one goes to stage three, etc. until payoff appropriation is possible. The concluding part of the argument is as before.

It is thus left to analyse maximization $\sum_{t \geq 1} \beta^t V(k_t)$ over the set defined by (14). We shall show that the sequence $k_t = k^* \; \forall t \geq 1$ maximizes the given function on a larger set, namely that it solves the problem

$$\begin{aligned} &\max_{\{k_t\}_{t\geq 1}} \sum_{t\geq 1} \beta^t V(k_t) \\ \text{subject to } &\sum_{t\geq s} \beta^{t\text{-}s}\, V(k_t) - D(k_s) \geq 0,\ s\geq 1\,. \end{aligned}$$

To this end observe that to improve upon the choice $k_t = k^*$ $\forall t \geq 1$ one has to raise at least one k_t from k^*. We show that this cannot be done without violating some constraint (keep in mind that if $k_t = k^*$ $\forall t \geq 1$ all the constraints hold with equality). Without loss of generality suppose k_1 is raised, say to $k^* + h_1$. By definition of k^*, cfr. equation (8), it will be $\Delta D > (1-\beta)^{-1}\Delta V$, so if one keeps $k_t = k^*$ $\forall t \geq 2$, since

$$\Delta\left(\sum_{t\geq 1} \beta^{t-1} V(k_t)\right) = \Delta V < (1-\beta)\Delta D < \Delta D$$

the constraint at $s = 1$ is violated (Δ refers here to raising k_1 from k^* to $k^* + h_1$ of course); hence to restore it one should raise k_2 —or some other k_t $t \geq 2$, the argument does not change. But by the same token, if one raises k_2 one then has to raise k_3 (or $k_{t_3}\ldots$) to restore the $(s = 2)$–constraint, and so on: that is, if k_1 is raised from k^* one should keep raising k's farther and farther away. Can this be done ad infinitum? The question here is, by how much does k_2 need to be raised to restore feasibility at $s = 1$? From the above displayed inequalities it follows that the needed increment of k_2 would be larger than the increment needed if the first inequality there were instead an equality, i.e. the h_2 such that

$$\beta\big(V(k^* + h_2) - V(k^*)\big) = \beta\big(D(k^* + h_1) - D(k^*)\big).$$

But $V(k^* + h_2) - V(k^*) < V'(k^*)h_2$, $D(k^* + h_1) - D(k^*) > D'(k^*)h_1$, and from (8) one has $V'(k^*) < (1-\beta)D'(k^*)$; therefore

$$h_2 > \frac{D'(k^*)}{V'(k^*)}\, h_1 > (1-\beta)^{-1}\, h_1 > h_1.$$

Analogously, to restore feasibility at $t = 2$ one should then have to raise k_3 by an amount $h_3 > h_2$, and by so doing it is clear that one hits the upper bound K in a finite number of steps. The conclusion is that it is in fact impossible to improve upon the choice $k_t = k^*$ $\forall t \geq 2$, as was to be shown.

Proof of Proposition 3. We first put on record an observation:

Lemma. *Fix a time τ, and assume $k_t = K_t$ $\forall t > \tau$. Then all incentive constraints for $s > \tau$ are satisfied with equality if $\phi_t = \hat{\phi}_t$ $\forall t > \tau$.*

Proof. Recall that by assumption (9) $V_t(K_t) = V_0(K_0)$, which in turn implies $V_t(K_t) = (1-\beta)\mathcal{V}$ $\forall t$. Then for $s > \tau$, given that $k_t = K_t$ and $\phi_t = \hat{\phi}_t$ for $t \geq s$, the incentive constraint at s is $(1-\beta)\mathcal{V}\sum_{t\geq s}\beta^{t-s}\hat{\phi}_t \leq \mathcal{V} - D_s(K_s)$, which, by assumption (10) and the fact that $\hat{\phi}_t = \gamma^{(s-\tau)+(t-s)}\hat{\phi}_\tau$, can be written as

$$\frac{(1-\beta)\mathcal{V}}{1-\beta\gamma}\,\gamma^{s-\tau}\hat{\phi}_\tau \leq \gamma^s\big(\mathcal{V} - D_0(k_0)\big).$$

We just have to plug in the definition of $\hat{\phi}_\tau$, page 9 to verify that equality holds. □

Consider now the case $t_0 = 0$, i.e. $\hat{\phi}_0 \leq 1$. Start again from observing that the incentive constraint at $t = 0$ has the objective function on the left. As before, try to maximize the right member and subject to have equality in the constraint. Since the right member is $\sum_{t\geq 0}\beta^t\, V_0(k_t) - D_0(k_0)$, set first $k_t = K_t$ $\forall t \geq 1$, and then $\phi_t = \hat{\phi}_t \ \forall t \geq 1$ to have the other constraints satisfied (with equality, from the lemma). This way the constraint at zero becomes

$$\phi_0 V_0(k_0) + \beta\big(\mathcal{V} - D_1(K_1)\big) \leq V_0(k_0) - D_0(k_0) + \beta\mathcal{V}.$$

If we set $k_0 = K_0$ and $\phi_0 = \hat{\phi}_0$ we have equality by definition, so from $\hat{\phi}_0 \leq 1$,

$$V_0(K_0) + \beta\big(\mathcal{V} - D_1(K_1)\big) \geq V_0(K_0) - D_0(K_0) + \beta\mathcal{V}. \tag{15}$$

Suppose first that $\hat{\phi}_0 < 1$, so that the above inequality is strict; if we lower k_0 from K_0 towards $\operatorname{argmax}(V_0 - D_0)$ the right member goes up, the left one down, and two possibilities arise: (i) equality is reached at some $k^* \in \big(\operatorname{argmax}(V_0 - D_0), K_0\big)$; in this case the value $V_0(k^*) - D_0(k^*) + \beta\mathcal{V}$ is the highest possible principal's payoff, attainable with $\phi_0 = 1$ (if we lower k_0 further the left member, i.e. the principal's payoff, decreases, with $\phi_0 = 1$ and even more for any $\phi_0 < 1$); thus optimal policy is here $k_0 = k^*$ (defined by the equality $D_0(k) = \beta D_1(K_1)$), $k_t = K_t \ \forall t \geq 1, \ \phi_0 = 1, \ \phi_t = \hat{\phi}_t \ \forall t \geq 1$; (ii) at $\operatorname{argmax}(V_0 - D_0)$ inequality in (15) is still strict; in this case the maximum possible value of $\sum_{t \geq 0} \beta^t \, V_t(k_t) - D_0(k_0)$, i.e. $\max\big[V_0 - D_0\big] + \beta\mathcal{V}$, is attainable with the $\phi_0 < 1$ defined by $\phi_0 V(\operatorname{argmax}(V_0 - D_0)) + \beta\big(\mathcal{V} - D_1(K_1)\big) = \max\left[V_0 - D_0\right] + \beta\mathcal{V}$, and optimal policy has the ϕ_0 just defined, $k_0 = \operatorname{argmax}\left[V_0 - D_0\right]$, and continuation for $t \geq 1$ as in the previous case.

If on the other hand $\hat{\phi}_0 = 1$, so (15) is an equality, then lowering k_0 from K_0 can only do harm (period–zero incentive constraint would hold with strict inequality, and the left member, i.e. the principal's payoff, would be lower than with $k_0 = K_0$). Hence optimal poicy in this case is $k_t = K_t \ \forall t \geq 0, \ \phi_0 = 1, \ \phi_t = \hat{\phi}_t \ \forall t \geq 1$. This ends the case $t_0 = 0$.

We now turn to the case $t_0 > 0$; recall that this means $\hat{\phi}_{t_0} \equiv \gamma^{t_0}(1 - \beta\gamma)(\mathcal{V} - D_0(K_0))/(1 - \beta)\mathcal{V} \leq 1$, but $\hat{\phi}_t > 1 \ \forall t < t_0$. Let us write the principal's payoff as

$$\sum\nolimits_{t=0}^{t_0-1} \beta^t \phi_t V_t(k_t) + \beta^{t_0} \sum\nolimits_{t \geq t_0} \beta^{t-t_0} \phi_t V_t(k_t). \tag{16}$$

Taking into account the constraints for $t \geq t_0$, which still have to be met, we know from the previous case what the constrained maximum of the second sum is, and the policy which achieves it.

Suppose first that $\hat{\phi}_{t_0} = 1$. Then if $k_{t_0} = K_{t_0}$, the incentive constraints for $t \geq t_0$ are all satisfied with equality (those for $t > t_{t_0}$ from the lemma, the one at t_0 checked easily). Moreover, in this case we shall now verify that it is feasible to set $k_t = K_t, \ \phi_t = 1 \ \forall t < t_0$; since this is the best one can hope for, optimal policy is found: $k_t = K_t \ \forall t$, $\phi_t = 1 \ \forall t \leq t_0$ and $\phi_t = \hat{\phi}_t \ \forall t > t_0$. To verify feasibility of the policy for $t < t_0$ consider the constraint at $t_0 - 1$, which with the given policy becomes

$$V_{t_0-1}(K_{t_0-1}) + \beta \sum\nolimits_{t \geq t_0} \beta^{t-t_0} \phi_t V_t(K_t) \leq \mathcal{V} - D_{t_0-1}(K_{t_0-1});$$

but $\sum_{t \geq t_0} \beta^{t-t_0} \phi_t V_t(K_t) = \mathcal{V} - D_{t_0}(K_{t_0}) = \gamma^{t_0}(\mathcal{V} - D_0(K_0))$, so the left member is equal to $(1-\beta)\mathcal{V} + \beta\gamma^{t_0}(\mathcal{V} - D_0)$; and since the right member is $\gamma^{t_0-1}(\mathcal{V} - D_0(K_{t_0}))$, the constraint is $(1-\beta)\mathcal{V} + \beta\gamma^{t_0}(\mathcal{V} - D_0(K_0)) \leq \gamma^{t_0-1}(\mathcal{V} - D_0(K_{t_0}))$; rearranging, this is just $\hat{\phi}_{t_0-1} \geq 1$, true by hypothesis. Analogously, the $(t_0$–$2)$–constraint becomes $\hat{\phi}_{t_0-2} \geq 1$, and so on down to zero.

Consider now the case $\hat{\phi}_{t_0} < 1$. Here as we know the policy maximizing the second sum in (16) calls for $k_{t_0} < K_{t_0}$, and this creates a trade–off: for a lower k_{t_0} implies a lower V_{t_0}, and this in turn tightens the incentive constraints for $t < t_0$. For example, it makes the policy of full transfer knowledge and full appropriation for $t < t_0$, found optimal just above when $\hat{\phi}_{t_0} = 1$, generally unfeasible. Given that optimal policy for $t > t_0$ remains the one defined before ($k_t = K_t, \ \phi_t = \hat{\phi}_t$), choice for $t \leq t_0$ solves the finite–dimensional problem just introduced, of maximizing $\sum_{t=0}^{t_0} \beta^t \phi_t V_t(k_t)$ subject to the constraints for $t \leq t_0$ (the values of k_t, ϕ_t for $t \geq t_0$ being fixed). This ends the proof.

Appendix B: Notes on the Linux Project

Some information is collected here about the economic history of Linux, in the hope that some readers will find it useful; it is also used in the concluding section 5. After mentioning the legal regime under which the system works we shall briefly go into past history and current situation.

Legal Underpinning: the GPL. The legal twist that gave birth, in the late eighties, to Open Source software is the *General Public Licence*, whose original idea is owed to the MIT programmer Richard Stallman. As all licences, the GPL and the several less 'radical' variants which are around by now are written by lawyers and for lawyers; we shall pass on the little we understand about them. [10]

Open source means that the user must be able to 'see the source [code]', but there is more in the GPL; the essential twist is, in Stallman's original wording, to 'turn copiright into copyleft': whereas copyright contains restrictions to use, modify and distribute a product, copyleft contains the restriction to restrict those things. [11] It is interesting that the GPL does not restrict the right to *sell* the programs covered by the licence —there is no need to. As Stallman puts it (`www.gnu.org/philosophy/selling.html`),

> "[there are] no requirements about how much you can charge for distributing a copy of free software. You can charge nothing, a penny, a dollar, or a billion dollars. It's up to you, and the marketplace, so don't complain to us if nobody wants to pay a billion dollars for a copy."

Indeed nobody will, because the impossibility to restrict redistribution induces competition between buyer and seller: buyer cannot make money by reselling product because if it tries to sell it for p the original seller can sell it for $p - \epsilon$; hence the buyer will not pay more than redistribution cost, so that the original seller will not make money in the first place.

Thus the rule underlying open source software production is in essence that every user (potentially user/developer) has the right to see what the users before her have done, and must pass this right on to the subsequent users. Of course, thinking of Linux again, this is surely not sufficient to generate a process of such import; there are motivational issues (why do programmers contribute to an open source project if they cannot 'make money'?), and problems of coordination and technical feasibility in the way, and we now turn to these.

Past. We touch upon three points: *(i)* the actors' motivations, *(ii)* organization, and *(iii)* technology of the Linux project.

(i) Motivations. About his decision to make Linux freely available, his creator Linus Torvalds says "[it] wasn't some agonizing decision that I took from thinking

[10] The two main sites about the topic are `www.gnu.org` and `www.opensource.org`, and contain all licences and extensive discussions. Useful for general understanding is in particular the commonly accepted Open Source *Definition*, which contains the essential requirements which open source licenses should satisfy. The idea of a definition which would serve as basis for the licences originates with Bruce Perens; his original version is in the Articles section of his web site, `www.perens.com`, of independent interest.

[11] "The central idea of copyleft is that we give everyone permission to run the program, copy the program, modify the program, and distribute modified versions —but not permission to add restrictions of their own. Thus, the crucial freedoms that define 'free software' are guaranteed to everyone who has a copy; they become inalienable rights." Stallman, at `www.gnu.org/gnu/thegnuproject.html`.

long and hard on it: it was a natural decision within the community that I felt I wanted to be a part of." (interview in First Monday, [7]). That crucial decision was effectively the only one available to him at the time, as Linus himself declares in his book [25] (chapter 2, section IX); but the point remains that it was a natural choice. For that community he wanted to be a part of was a community to which all members were feeling good to belong, in a somehow deep sense. It was like when hippies liked to be hippies in the States, or the '68–guys liked to be what they were in Europe; and more and more all those involved perceived to be part of something great and important (at least this is the impression I get by watching the hackers' community from outside). Viewed in this light the economics question "How come all these guys have contributed apparently for free" sounds stupid at first sight; but of course it must be read as "What is there *besides* social magic behind their motivations". [12] First there is some 'individual' magic, like there is in mathematics; Torvalds, for one, thinks this —'the fun'— is the main thing (Torvalds [25]). [13] Then there are two purely economical forces: the signalling motive, and the direct use–value to the user–developer of his own contribution. On the first, the insiders' view is not difficult to guess; see e.g. Raymonds [21], or Torvalds in [7]; critical is also Benkler [3]. [14] User value on the other hand has certainly been decisive for the major open source projects (see e.g. Lerner–Tirole [16]). And it is worth noting that even outside the software industry, the relevance of user–driven product development is widely recognized; see most notably von Hippel [26, 27, 28]. Also, to the individual use value one must add the cumulative effect of concurring contributions —as Ganesh Prasad puts it for software development, "Each programmer contributes a brick and each gets back a complete house in return." [15] This picture of motivations will be recalled when we comment on replicability of the open source model outside software production in the concluding section.

(ii) Organization. [16] The hierarchical organization of the Linux project (and of most of its satellite projects) is usually of a 'benevolent dictatorship' (cfr. Dafermos

[12]Concerning 'who contributed what', in their survey on 13,000 contributors to open source projects Ghosh and Prakash [9] found that three quarters made only one contribution, but at the other end nearly three quarters of contributions came from the top ten percent of contributors. I suppose the social excitement factor alone is enough to explain the one–timers' contributions; the question is really about the hard–working guys. Note that even for the latter the social factor is not irrelevant, for they were leaders of a large generational movement, and obtaining and maintaining such a position may well be worth a lot of hard work.

[13]In fact 'fun' is more for him: it is the third and last of the three stages of evolution of humanity according to his (non–trivial) theory of evolution, the first two being survival and social order; see [25].

[14]Torvalds, after much pressure from the interviewer responds "Yes, there are issues involved with 'getting value back' from your involvement [...] but the first consideration for anybody should really be whether you'd like to do it *even if* you got nothing at all back". Benkler adduces the fact that some of the most important projects, like the Apache Web Server and the Free Software Foudation, do not provide personal attribution to the code they produce. In fact much more is true, cfr. Ghosh–David [8]: in the Linux kernel consistently more than half of the code is unsigned; and those packages whose lines of code are entirely signed constitutes the 0.66% of total kernel packages.

[15] `http://linuxtoday.com/infrastructure/2001041200620OPBZCY--`.

[16]In the sense of *industrial* organization. Many have written about the organization of the project from a *social* point of view. Not surprisingly the parallels to the academic model of open knowledge production are ubiquituos: besides Raymond [21] see e.g. Himanen [11], who starts from Plato's Academia, and Benkler [3] who is himself an academic. From an evolutionary point of view, interesting is the essay by Kuwabara [15] who, based on the gift–culture idea of Raymond, gives an interpretation of the process in the light of the the Santa–Fe approach to the dynamics of complex systems. A link to the work of J.B. Arthur from Santa–Fe comes also from Dafermos [6] in connection with increasing returns.

[6]); the dictator must also be *trusted*, and trust is conferred by public legitimacy. As to productive organization, the clearest insight for understanding emergence of 'peer production' comes in my opinion from Benkler [3] (NYU School of Law). Benkler makes a conditional statement, *given* strong enough actors' motivations (cfr. above) and technological feasibility (on this shortly). With these assumptions in place, to the two dimensions of transaction costs/organization costs responsible for the firm/market tradeoff in Coase's theory Benkler adds a third dimension: that of 'information opportunity costs', and correspondingly a third alternative mode of production: peer production. The idea is that the latter may prevail due to the advantages which decentralized information gathering and exchange gives in identifying and allocating creative work to the more appropriate jobs.

(iii) Technology. The essential characteristics of the process are well understood: there is a substantial initial 'core' of potential widespread use (Weber [29]); the subsequent product development is modular (also Benkler [3] and Lerner–Tirole [16]); the size of the modules is small ([3], [16]; Benkler uses the term 'granularity'); and modules integration (quality control and decision processes) is managed effectively (cfr. [3]). Lerner–Tirole argue that lack of granularity is the main technological obstacle to transposition of the open source/peer production model to other industries: "[. . .] In many industries, development of individual components require large team work and substantial capital costs" ([16], p. 231). This is undoubtedly true, but not necessarily pervasive; von Hippel [28] for instance cites old empirical research on technological innovation (concerning the post–war decades) showing that both in Rayon manufacture and computer hardware the cumulation of a multitude of minor technical changes is "responsible for much or most technical progress" ([28] p.14). Distinguishing the different stages of product maturity, is seems reasonable to expect large teams and investments more frequently needed in an initial phase, followed by a cumulation of minor improvements taking place in a subsequent one.

Present. In the last couple of years much has changed. Ghosh and Prakash [9] found in 2000 that 75% of contributors to open source projects were one–timers; there are no surveys about the current situation yet, but the obvious guess is that the times of one–timers have gone (just read on). We will talk separately about *(i)* for–profit firms commercializing Linux, and *(ii)* the *Open Source Development Labs.*

(i) Red Hat & C. In a paper appeared in August 2003, Haruvy et al. [10] solved what would have been Red Hat's optimal control problem with commercializing open source software in a situation like the one of the 2000 Survey, namely: if such a firm charges a short–run profits maximizing price, the high profit realized may induce spite in the hackers' community, hence decreased contributions, hence lower future product quality, hence ultimately lower profits. Well, forget it. Remember the Red Hat Box selling at around 50 US Dollars? In mid–October 2003 it was still selling, but it was pretty hard to find in their web site (at least for me); by end–October, it had disappeared completely! The point is that it is not what they sell any more; they sell Red Hat 'Architecture', and Red Hat 'Solutions', to firms who become Red Hat customers. No role for hackers left to play. Open Source, Linux–based software is now getting ready to replace proprietary (mostly Unix–based) software infrastructure at mission–critical level, in the communication market. The operating system is just a part of a much more complex product, and competition is growing between 'Red Hat Architecture' against 'IBM Linux Solutions', 'SuSe for the Enterprise', etc. (incidentally Red Hat and VA Software, the two leading firms of the sector quoted at Wall Street, have seen their stock value more than doubled in the last three months, August–October 2003).

(ii) The OSDL. What about Linus, who started it all? Miniaturized along with old kernel problems? Contrarily to what one could expect, the answer is no — indeed, all the opposite. To migrate to Linux in corporate data centers and in telecommunications networks, the interested companies want reassurance that the system meet some critical 'carrier-grade' requirements. And since different firms have different technologies and priorities, sofware developers on their part need to know what exactly these requirements are, and how they are ranked in terms of priority. So the crucial step becomes the creation of a 'focal' set of requirements *definitions* and priority ranking (of course evolving with time), based on inputs from the business sector and to which developers can refer for their programming objectives. Supported by a global consortium of Information Technology industry leaders, the *Open Source Development Labs*, a non-profit organization, was founded in 2000 for exactly that purpose. [17] Most big corporations were involved, but the vital 'authority', in the sense of recognition from the community the way Torvalds had been for the early kernel development, was missing. What exactly was missing is easily guessed: Linus in person, of course. Well, since June 2003 that is where he is: in this new Linux world, again at what is becoming an important gravitational center of it. In essence, in the market sketched sub *(i)* above, where product complexity makes the source more and more 'hidden', the role if OSDL is that of keeping the core of it common and open.

References

[1] Aghion, Philippe and Jean Tirole (1994): The Management of Innovation, *The Quarterly Journal of Economics* **109**: 1185–1209

[2] Barbu, Viorel and Theodor Precupanu (1986): *Convexity and Optimization in Banach Spaces*, D. Reidel Publishing Co.

[3] Benkler, Yochai (2002): Coase's Penguin, or, Linux and 'The Nature of the Firm', *Yale Law Journal* **112**

[4] Bessen, James and Eric Maskin (2000): Sequential Innovation, Patents and Imitation, MIT Working Paper no. 00-01

[5] Birch, Steve (2003): R&D Outsourcing Strategies, *Pharmafocus* (July), online at `www.pharmafile.com/pharmafocus`

[6] Dafermos, George N. (2001): Management and Virtual Decentralised Networks: The Linux Project, *First Monday* **6** No.11

[7] First Monday (1998): Interview with Linus Torvalds: What motivates free software developers? *First Monday* **3** No.3

[8] Ghosh, Rishab Aiyer and Paul A. David (2003): The nature and composition of the Linux kernel developer community: a dynamic analysis, *Stanford Project on Economics of Open Source Software*, mimeo Stanford University

[9] Ghosh, Rishab Aiyer and Prakash, V.V. (2000): The Oribiten Free Software Survey, 1st Edition, online at `http://orbiten.org`

[10] Haruvy, E., A. Prasad and S.P. Sethi (2003): Harvesting Altruism in Open–Source Software Development, *Journal of Optimization Theory and Applications* **118**: 381-416

[11] Himanen, Pekka (2001): *The Hacker Ethic and the Spirit of the Information Age*, Random House

[12] Holmström, Bengt and John Roberts (1998): The Boundaries of the Firm Revisited, *Journal of Economic Perspectives* **12**: 73–94

[13] Kimzey, Charles H. and Sam Kurokawa (2002): Technology Outsourcing in the U.S. and Japan, *Research Technology Management* **45**: 36–42

---

[17] The web site is `www.osdl.org`. The consortium includes from hardware producers like Cisco, Dell, HP, IBM, Intel, Mitsubishi, Nec, Sun and Toshiba, to firms involved in telecommunications like Ericsson and Nokia, and more software oriented firms like Linuxcare, Red Hat, SuSe, Turbolinux. The directory `lab_activities/carrier_grade_linux/documents.html` contains the main charter document, together with a 'Technical Scope White Paper'; the analogous White Paper for Data Centers is in `lab_activities/data_center_linux/documents.html`. The technical Requirements Definition (version 2.0) is of course also available at their site. A 'need to know' paper about carrier grade linux, written for engineers, is Mehaffey [17].

[14] Kultti K. and T. Takalo (2000): Incomplete contracting in an R&D project: the Micronas case, *R & D Management* **30**: 67–77

[15] Kuwabara, Ko (2000): Linux: a Bazaar at the Edge of Chaos, *First Monday* **5** No.3

[16] Lerner, Josh and Jean Tirole (2002): Some Simple Economics of Open Source, *Journal of Industrial Economics* **50**: 197–234

[17] Mehaffey, John (2002): Carrier Grade Linux: What You Need to Know, *Communication Systems Design*, online at `www.commsdesign.com/story/OEG20020827S0008`

[18] Mowery, David C. and Nathan Rosenberg (2000): *Paths of Innovation: Technological Change in 20th–Century America*, Cambridge University Press

[19] Rajan, Raghuram G. and Luigi Zingales (2001): The Firm as a Dedicated Hierarchy: A Theory of the Origins and Growth of Firms, *Quarterly Journal of Economics* **116**: 805–851

[20] Ray, Debraj (2002): The Time Structure of Self–enforcing Agreements, *Econometrica* **70**: 547–582

[21] Raymond, Eric S. (1998): Homesteading the Noosphere, *First Monday*, **3** No.10

[22] Rustichini, A. (1998): Lagrange Multipliers in Incentive–constrained Problems, *Journal of Mathematical Economics* **29**: 365–380

[23] Scotchmer, Suzanne (1999): Cumulative Innovation in Theory and Practice, GSPP Working Paper no. 240, UC Berkeley

[24] Thayer, Ann M. (1997): Outsourcing R&D to Gain an Edge, *Chemical & Engineering News* (February), American Chemical Society, online at `http://pubs.acs.org/hotartcl/cenear/970210/rd.html`

[25] Torvalds, Linus and David Diamond (2001): *Just for Fun: The Story of an Accidental Revolutionary*, Harper Business

[26] von Hippel, Eric (1986): Lead Users: a Source of Novel Product Concepts, *Management Science* **32**: 791–805

[27] von Hippel, Eric (1998): Economics of Product Development by Users: Impact of 'Sticky' Local Information, *Management Science* **44**: 629–644

[28] von Hippel, Eric (2002): Horizontal Innovation Networks —by and for Users, *MIT Sloan School of Management* WP No. 4366–02

[29] Weber, Steven (2000): The Political Economy of Open Source Software, BRIE Working Paper 140, Berkeley University