

Open source software development : some historical perspectives

Citation for published version (APA):

Nuvolari, A. (2003). *Open source software development : some historical perspectives*. (ECIS working paper series; Vol. 200301). Technische Universiteit Eindhoven.

Document status and date:

Published: 01/01/2003

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.



Open source software development: some historical perspectives

Alessandro Nuvolari

Eindhoven Centre for Innovation Studies, The Netherlands

Working Paper 03.01

Department of Technology Management

Technische Universiteit Eindhoven, The Netherlands

January 2003

Open Source Software Development: Some Historical Perspectives*

Alessandro Nuvolari
(Eindhoven Centre for Innovation Studies)

ABSTRACT

In this paper we suggest that historical studies of technology can help us to account for some, perplexing (at least for traditional economic reasoning) features of open source software development. When looked in historical perspective, open source software seems to be a particular case of what Robert Allen has termed “collective invention”. We explore the interpretive value of this historical parallel in detail, comparing open source software with two remarkable episodes of nineteenth century technical advances.

Keywords: Open source software – Collective Invention – Blast Furnaces – Steam engines – Intellectual Property Rights

Contact author: Alessandro Nuvolari, ECIS (Eindhoven Centre for Innovation Studies), Faculty of Technology Management, Eindhoven University of Technology, P.O. Box 513, 5600 MB, The Netherlands. E-mail: a.nuvolari@tm.tue.nl

* I would like to thank Nick von Tunzelmann, Richard Nelson, Bart Verspagen, Stefano Brusoni and Roberto Fontana for useful comments on a previous draft of this paper . Of course, controversial opinions and remaining errors are entirely my own.

1. Introduction

Open source software development has begun to draw the systematic attention of economists and social scientists alike. Two main reasons are at the heart of this upsurge of interest. The first one is the growing importance of open source products in the recent development of the software industry. A number of open source projects has been crowned with a remarkable technical and economic success. Apache (a web server), Linux (an operating system), Sendmail (an Internet transfer mail agent) are the most notable examples. The second reason is that some features of open source software development appear, at least at first sight, quite paradoxical to traditional economic reasoning. Lerner and Tirole (2001) individuate four “key research questions” that are in need of explanation:

1. As a spontaneously provided “pure” public good, open source software should be prone to the free-rider problem; how can open source software projects induce the active participation of talented developers who are not directly rewarded for their efforts ?
2. Relatedly, why do profit motivated firms collaborate in open source projects ? What type of economic return do they expect ?
3. The most notable cases of open source are highly complex software products – involving an articulated division of labour and the solution of a series of stringent coordination problems – how loosely coordinated networks of “hackers”¹ have been able to manage effectively this complexity ?
4. Finally, open source projects are based on an institutional arrangement (public license) in which the prerogatives descending from the creation of new useful knowledge are very different from those which are granted under traditional

¹ In the programmers’ community the term “hackers” is used to indicate those “who love to program and enjoy being clever about it” (Stallman (1999), p.53). The term “hackers” is very often used in the media to denote those who try to sabot computer systems. This use of the word, however, is completely unwarranted.

intellectual property rights regimes (patents, copyrights and trade secrets). What is the impact of this specific institutional arrangement on the rate of technological innovation ? How does it perform compared with traditional schemes intellectual property rights protection ?

In this paper we suggest that nineteenth centuries experiences of technical change can provide some useful insights for the investigation of these issues. When looked in historical perspective, open source software seems to be another case of a particular type of innovation process that Robert Allen has termed as “collective invention”.² Within “collective invention” settings, rival firms (or independent individual developers) freely release each other *pertinent* information concerning the solution of non trivial technical problems. Each firm, in turn, makes use of the received information to incrementally improve on a basic common technological layout. It seems worthwhile to dig deeper into this apparent historical parallel. We compare open source software development with two episodes of nineteenth century technical change. The first one is the case of the iron industry of Cleveland (UK) described in Allen’s paper, while the second one is the case of the Cornish pumping engine. In this way, we hope to get a deeper understanding of the salient features of open source software and to provide a preliminary interpretative framework that can fruitfully guide further research.

The rest of the paper is organised as follows. The next section provides a short account of the historical evolution of open source software development. Section 3 is devoted to a thorough re-examination of Allen’s paper. This is necessary because some of the conceptual issues raised in that article have been not fully appreciated in the subsequent literature. Section 4 describes the case of the Cornish pumping engine. This case is particularly interesting for our purposes because it originated from a harsh dispute on intellectual property rights that has striking resemblances with the ongoing conflict between the open source community and Microsoft. The final section summarizes and draws conclusions.

² Allen (1983).

2. Open source software: a short interpretive history

Open source programs are computer programs which are distributed together with the source code.³ This makes possible for sophisticated users to modify the code and introduce improvements and/or modifications in the programs. In turn, additions and modifications to open source programs are also redistributed together with their source code. As a consequence, open software projects tend to involve a fairly large number of different developers, whose main “connecting agent” is constituted by the mere sharing of the source code. The actual details of the distribution can vary with the specific license under which the programs are distributed: some programs are available for free, others are normally sold for a price.

The practice of sharing the source code of programs is not an entirely novel feature of the software industry. The first “intensive” users of mainframe computers were universities and corporate research laboratories. In that environment, computer programs were eminently seen as research tools, which was normal to share with other developers. Richard Stallman describes what was the typical practice at the MIT Artificial Intelligence laboratory in the early 1970s in these terms:

Whenever people from another university or a company wanted to port and use a program, we gladly let them. If you saw someone using an unfamiliar and interesting program, you could always ask to see the source code, so that you could read it, change it, or cannibalize parts of it to make a new program.⁴

This practice of sharing programs among users generated a powerful drive towards the creation of so-called “portable” software (that is software that could easily be ported on different computer platforms). A major step in this direction was the development of the Unix operating system (by Ken Thompson) and of the C programming language (by Dennis Ritchie) at Bell Labs

³ The source code of programs is written in computer language such as (C, Java, Pascal, etc.) . Proprietary software producers typically distribute only the object code (the series of 0s and 1s, which is actually read by the machine).

⁴ Stallman (1999), p. 53. Cooperation in software development was also enhanced by the Arpanet network (a network connecting mainframe computers at universities, research labs and other defence contractors) in 1969 (Raymond,1999, p. 8)

in 1969. Unix could be run on a wide range of machines.⁵ Additionally, since 1979, Unix machines had also started to be connected in a network. The creation of a community of interconnected Unix users stimulated further the habit of sharing programs.⁶

This situation changed dramatically in the early 1980s. AT&T (not anymore legally constrained to its role of telephone company) began to sell licenses on Unix. Furthermore, concomitant advances in computer technology (the widespread diffusion of the PC and of the workstation) reinforced the drive towards the increased commercialization of software products.⁷ Many talented programmers moved away from universities and research labs to private software firms, where they were bound by non disclosure agreements. In the case of Unix, the proliferation of different commercial versions proved to be disastrous, leading to the balkanisation of the users community and progressively frustrating the promise of a generalized cross-platform portability of the applications.⁸

In reaction to these developments, in 1984 Richard Stallman (a programmer previously employed at the MIT Artificial Intelligence laboratory) founded the Free Software Foundation (FSF). The aim of the foundation was to contribute to recreate the “open” environment characteristic of the early years. The first task chosen by the FSF was the production of a non proprietary operating system in order to create an “open” environment in which non proprietary programs could run. This operating system, which was supposed to be endowed with the features of Unix, was named GNU (which stands for “GNU is Not Unix”).

In the late 1980s, Stallman and the FSF recreated a non proprietary “GNU version” of many components of the Unix software. The programs were developed in such a way that they could run on almost every version of Unix. The development of GNU software was organized by

⁵ On the early history of Unix see Salus (1994).

⁶ This was permitted by the development of a new feature of Unix, the UUCP (Unix-to-Unix-Copy), which made possible to exchange files and data using telephone lines.

⁷ On the evolution of the US software industry in the 1980s, see Steinmueller (1996), pp. 30-41. The growth of the software industry in the United States was also accompanied by a progressive strengthening of legal protection for software in the US (until the upholding of patents for “pure” software products). See Merges (1996).

⁸ Raymond (1999), p. 22.

means of a sort of “future tasks” list, which was used to stimulate FSF collaborators to work at the development of the missing parts of the GNU system.⁹

The FSF project also included the creation of a Unix compatible kernel,¹⁰ called Hurd. The development of the kernel proved to be the most difficult stage of the GNU project. Stallman and his group seemed to be overwhelmed by the technical difficulties of the undertaking, so that the release of Hurd was repeatedly procrastinated.¹¹

In order to protect the GNU software from being turned into proprietary software, Stallman introduced a particular licensing procedure called General Public License (GPL, also known as “copyleft”). The GPL permits the free distribution, modification and redistribution of the modified version of the programs it covers. The main characteristic feature is that modified versions of programs licensed under the GPL, must be also licensed under the same terms. This is also called the “viral” clause, because it “infects” all the code that is bundled together with GPL pieces of code.

The creation of a non proprietary Unix-like kernel was the result of a rather unexpected development. In 1991 Linus Torvalds, a computer science student at the University of Helsinki, announced on an Internet newsgroup, that he was working on a free version of Unix and he asked for help in bug fixing. Torvalds also declared that he was willing to include in future versions, new features developed by others as long as they would have also been freely redistributable (Torvalds adopted the GPL-copyleft license scheme).

The initiative met an extraordinary success . In 1994, when Torvalds released Linux 1.0, the operating system could compete successfully in stability and reliability with commercial versions of Unix. An interesting feature of the distribution scheme adopted by Torvalds is that users can chose between even-numbered releases which are relatively more stable and bug-fixed, and odd-numbered releases which incorporate the latest developments and for this reason have a more

⁹ Stallman (1999), p. 62.

¹⁰ The kernel is the core part of the operating system that controls access to hardware components.

¹¹ In 1999 Stallman declared that Hurd is not yet ready for practical use (Stallman (1999), p.63).

experimental nature. When the current odd-numbered version is sufficiently tested and bug-fixed, it is then distributed as the next even-numbered.

In the second half of the 1990s, Linux was further refined, incorporating a number of new features. The community of the developers grew exponentially. The decisive “official recognition” of its potential came, perhaps, in 1999, when an internal memorandum which leaked out from Microsoft, individuated in Linux (and, in more in general terms, in the diffusion of the open source process of software production) as a major competitive threat for the company.¹²

Eric S. Raymond (who is himself an hacker involved in several “open source” projects), has attempted to outline a first comprehensive appraisal of the “open source” phenomenon (Raymond, 1999). In a famous paper titled “The Cathedral and the Bazaar”, Raymond deals with the problems arising from the management of large and complex software projects. He draws a contrast between two archetypical modes of software development, which he labels the “cathedral” and the “bazaar”. In the cathedral mode, software is developed from a unified a-priori project that prescribes all the functions and the features to be incorporated in the final product. Programmers’ work is centrally coordinated and supervised, in order to assure the integration of the various components. Needless to say, this mode of development is characteristic of commercial software. In opposition to the “cathedral”, there is the “bazaar” mode of development, where software emerges from an unstructured evolutionary process. Starting from a minimal piece of code, groups of uncoordinated programmers proceed by adding features and introducing modifications and patches to the code. There is no central allocation of the different tasks, but developers are left completely free to make the program evolve in the direction they favour. This is, what seems to be, the Linux style of software development. Amazingly, the bazaar mode of software development seems, to work and to be able to produce effectively highly complex software products.

¹² The report is known as the “Halloween document”. See Vallopillil (1998). On the evolution of Linux in second half of the 1990s see Moody (2001).

Open source projects are typically characterized by a very loose degree of coordination.

Decentralized developers are completely free to introduce the modifications or corrections they wish to the program. The “owner” of the project (that is the person who is in “charge” of the project and who releases the successive “official” versions of the program) - will check them and integrate the most valuable ones in future releases. However, Raymond, in his analysis, clarifies that this type of “bazaar” development is based on a necessary *precondition*.

It is fairly clear that one cannot code from ground up in bazaar style. One can test, debug and improve in bazaar style, but it would be very hard to *originate* a project in bazaar mode.....Your nascent developer community needs to have something runnable and testable to play with. When you start community building, what you need to be able to present is a *plausible promise*. Your program doesn't have to work particularly well. It can be crude, buggy, incomplete and poorly documented. What it must not fail to do is (a) run, and (b) convince potential co-developers that it can be evolved in something really neat in the foreseeable future. ¹³

Raymond's “plausible promise” defines an essential feature of a successful open source projects.

Raymond points out that, although an open source project is always in a somewhat a *fluid design state*, nevertheless it should define, possibly from the very outset, the broad lines of a solid architectural structure capable of sustaining the streams of future decentralized developments.¹⁴

In his subsequent works, Raymond has also addressed the issue of the incentive structure underpinning open source projects. It is useful to approach the problem by considering the nature of programming activity. Programming is a creative problem-solving activity. The creative element introduces into it a sort of aesthetic dimension, so that one can speak of “elegant” or “beautiful” solutions, as well of innovative solutions. For example, speaking about Linux, Linus

Torvalds observes:

Originally Linux was just something I had done and making it available was mostly a “look at what I've done – isn't this neat ?” kind of thing. Hoping it would be useful to somebody, but certainly there is some element of “showing off” in there too. ¹⁵

¹³ Raymond (1999), pp. 57-58 (italics in the text).

¹⁴ The Unix-type architecture is particularly suitable of “modularized” developments. Modularized developments are probably also enhanced by the fact that the “common ethos” of open source communities tend explicitly to favor the submission of “neat” modifications or additions. See Moody (2001, p.14)

¹⁵ Torvalds (1999).

The value of a programming contribution, then, can be “properly” appreciated only by other experts. In open source projects, individual contributions are acknowledged in a credit file. Thus, participants in open source projects are provided with the appreciation of a large competent audience. The desire for independent peer-recognition, according to Raymond, represents one of the most important individual incentives for getting involved in open source projects.¹⁶

In this respect, it is important to notice that participants’ belief that their contribution will be fairly appraised and, if deserving, taken into account in future releases is of outmost importance for the success of the project. This points to another necessary precondition for the success of an open source project: the legitimisation of the “owners” of the project. Participants must trust the capacity of the owner, not only of judging the merits and the limits of every contribution, but also his ability of merging them (avoiding conflicts that might ultimately lead to “forking”) in such a way to assure that the overall project will evolve in a coherent way.

The reward system based on peer review system used in open source software projects seems to mimic quite closely procedures that are typical of scientific research. Dasgupta and David (1994) have argued that the fundamental difference between science and technology consists in the different institutional set-ups that “regulates” the production of knowledge in the two domains. Scientific research is governed by a reward system based on public disclosure of the findings and peer-review, whereas the production of “technological” knowledge tends to have a distinctively proprietary character, being protected by patents or trade secrets (the rewards for the production of this type of knowledge typically derive from the “commercialisation” of the fruits of the research).

¹⁶ Lerner and Tirole (2002) argue that open source projects act as a stage where programmers can signal their skills and enhance their future career perspectives. Raymond also notices that “Occasionally the reputation one gains in the hacker culture can spill over into the real world in economically significant ways”, however, in his view, this is rather sporadic so it would be wrong to regard it as the main motivational drive (Raymond (1999), p.97)

Expanding on the work by Dasgupta and David, Foray (2000) has noticed that, *in specific historical instances*, also the production of technological knowledge appears to have been governed by sets of “open knowledge institutions”, clearly akin to the ones of “open science”. According to Foray, open source software development is one of these cases. Another example put forward by Foray in his paper is the case of the Cleveland (UK) iron industry described by Robert Allen (1983). In the remaining, we will explore the historical parallel proposed by Foray in detail, trying to derive further insights for the interpretation of the procedures of “open source” software development. Before doing that, it is important to remark that, although software creation is the result of a purely intellectual production process, it does share important similarities with other more “mundane” industrial engineering activities, in particular with those dealing with the production of “complex products” . Complex capital goods like complex software systems are composed of a large number of interacting components. In these cases, the outcome of all the potential interactions cannot be fully predicted *ex-ante*. Thus, a number of defects and limitations can be identified and “debugged” only after a, more or less long, phase of actual use of the product in question.¹⁷

3. Collective invention: the Cleveland blast furnaces.

According to Allen, in capitalist economies four main sources of invention can be discerned: (1) non-profit institutions such as universities a government financed research centres, (2) private firms R&D laboratories, (3) individual inventors, (4) collective inventions settings. In collective invention settings, competing firms freely release each other information on the design and the performance of the technologies they have just introduced. Allen has noticed this pattern of behaviour in the iron industry of Cleveland (UK) during the mid-nineteenth century:

...[I]f a firm constructed a new plant of novel design and that plant proved to have lower costs than other plants, these facts were made available to other firms in the industry and to potential entrants. The next firm constructing a

¹⁷ The similarity between software and the construction and implementation of complex engineering products is also suggested in Rosenberg (1982, pp 120-149). Also, Torrisi (1998, p. 42) emphasizes that some procedures of software production are pretty close to those of mechanical engineering.

new plant build on the experience of the first by introducing and extending the design change that had proved profitable. The operating characteristics of the second plant would then also be made available to potential investors. In this way fruitful lines of technical advance were identified and pursued.¹⁸

Information was normally released through both formal (presentations at meetings of engineering societies and publications of design details in technical journals) and informal channels. Additionally, the released information was not protected by patents. As a consequence of this information sharing, Allen shows that, in the period in question, the height of the furnaces and the blast temperature increased steadily by means of a series of small, but continuous rises. Increases in furnace height and blast temperature resulted in lower fuel consumption and in reduced production costs.¹⁹

As we have seen, the historical account outlined in the previous section suggests that open source software development can also be seen a case of “collective invention”. In his paper, Allen describes the very restrictive set of conditions that allowed such a pattern of technical change to emerge.²⁰ In his story, Allen points to the combined effect of three essential factors.

The *first* element has to do with the nature of technological advance. In the period in question, there was no consolidated theoretical understanding of the working of the blast furnace. Thus, the performance of a new blast furnace was to an important degree uncertain. The best engineers could do was to derive design principles on the basis of the previous experiences. In an analogous vein with the odd and even Linux versions, iron industrialists could commission either a new “leading edge” blast furnace (i.e. a taller furnace operating at higher temperatures) bearing

¹⁸ Allen (1983), p.2.

¹⁹ It is important to notice that Allen’s notion of “collective invention” does not refer to the exchange of information between users and producers studied by Lundvall (1988). “Collective invention” also differs from the “know-how trading” described by Von Hippel (1987). In “know-how trading”, engineers “trade” proprietary know-how in the sense the information is exchanged on a bilateral basis (non-participants to the transaction in question are excluded). Within collective invention, all the competing firms of the industry have free access to the potentially proprietary know-how. See Von Hippel (1987), pp. 296-297.

²⁰ Harhoff, Henkel and Von Hippel (2002) individuate, using a game-theoretic approach, a number of contexts in which “free revealing of innovations” is economically rational.

the related risks of a more uncertain future performance, or an imitation of what was deemed to be best existing “safe” design, which was, by far, a lower risky choice. Examples of both types of behaviour are documented.²¹

Secondly, Allen shows that a mechanism based on reputation, assuring the existence of a mutual individual incentive to disclose technical information to outsiders, was also at work in Cleveland. Blast furnaces were designed by independent consulting engineers who moved from firm to firm. The diffusion of technical information concerning the design and the performance of the different blast furnaces, allowed the engineers to consolidate their reputation and improve their future career prospects.

Thirdly, the disclosure of information did not prevent to the owners of the blast furnaces from reaping economic benefits from their innovations (so that one does not need to invoke any relinquishment of profit oriented behaviours in order to account for knowledge disclosure). Iron industry entrepreneurs, in most cases, were also owners (or they had mining rights) of the iron ore mines of the Cleveland district. The improvements in the efficiency of the blast furnaces of the district determined a substantial increase in the value of the iron ore deposit located in the same area.²² It is interesting to note, that the fact the iron firms were also owners of the iron ore deposit made them more interested in improvements of the *average aggregate performance* of the blast furnaces of the industry rather than in that of individual ones (only improvements in the average aggregate could actually determine increases of the value of the iron deposit). In economics terms, the receiver of the externality (information disclosure) by erecting a new well-performing blast furnace, in turn generated a new externality beneficial to the sender. This self-reinforcing process further stimulated the propagation of information among iron producers.

²¹ Allen (1983), p.5.

²² Because of the chemical composition of the iron ore and of particular factor prices circumstances, technical progress in Cleveland blast furnaces was location specific and other iron producing areas could not benefit from the innovations introduced there (Allen (1983), pp.17-19).

This pattern of behaviour has also a clear counterpart in the case of open source software, with the entrance of commercial companies in market segments that are *complementary* with open source software projects.²³ In this way, we see the emergence of an industry structure similar to Cleveland, where the returns from collective innovative activities are reaped via the development of complementary market segments that are positively affected by the rapid rate of innovation in the “knowledge sharing” sector. Companies like Red Hat, Caldera and Suse, for example, have a concrete interest in fostering the continuous improvement of Linux.²⁴ Hence, we see that these companies are effectively sponsoring new developments of Linux and other open source initiatives. The historical parallel, in this case, can explain why private companies operating in “complementary segments” are willing to release the results of their open source efforts (this can also explain why to these companies the alternative strategy of “polluting” the open source process developing proprietary software components does not appear as particularly promising).

4. The Cornish pumping engine²⁵

A particularly striking episode of nineteenth century technical change is the case of the Cornish pumping engine. The Cornish pumping engine fits nicely in Allen’s notion of collective invention. Furthermore, in the case of the Cornish pumping engine, a debate on alternative intellectual property rights regimes featured prominently. Thus, being intellectual property rights issues often raised in the ongoing discussion of the merits and limitations of open source software development, it is probably useful to examine this historical episode in detail.

In the seventeenth and eighteenth centuries mining activities were severely hampered by flooding problems. Not surprisingly, some of the first attempts at employing steam power were aimed at

²³ On the possible commercial software reactions to open source projects see Lerner and Tirole (2002) and Raymond (1999), pp 137-195.

²⁴ These companies sold “tested” versions of Linux together with a number of support services. The main source of revenues is represented by the sales of services. On the business strategy of Red Hat see Young (1999) and Raymond (1999, pp. 165-166 and 183-188)

²⁵ This section draws on Nuvolari (2002).

finding a workable solution to mine draining problems. In 1712, after a prolonged period of experimentation, Thomas Newcomen developed a steam pumping engine that could be used effectively for mine drainage. Using steam at only atmospheric pressure, the Newcomen engine was well within the limits of the engineering capabilities of the time. Moreover, the Newcomen engine was robust, reliable and based on a quite simple working principle. As a consequence, once it was installed, it could work for a long period with almost negligible maintenance costs. Given these merits, it is not surprising that Newcomen types of engines soon became of widespread use in mining activities.

The Newcomen engine had the major shortcoming of a high fuel consumption, which was determined by the necessity of alternatively heating and cooling the cylinder at every stroke. In coal mining, where large supplies of cheap coal were available, high fuel consumption did not represent a major limitation, but in other mining areas fuel inefficiency did not permit a widespread diffusion of the engine (von Tunzelmann, 1978, chap. 4).

Since the early diffusion of the Newcomen engine, fuel consumption was considered as the main “metric” to be used in the evaluation of the overall performance of a steam engine. The most common measure of fuel efficiency was termed the “duty” and was calculated as the quantity of water (measured in lbs.) raised 1 foot high per 1 bushel (84 lbs.) of coal consumed. From an engineering viewpoint, the duty is a measure of the thermodynamic efficiency of the steam engine. However, ‘duty’ has also an important economic meaning because it is a measure of the productivity of a steam engine with respect to the largest variable input used in the production process (von Tunzelmann, 1970, pp.78-79).

In 1769 James Watt conceived an alteration to the basic design of the steam engine (the introduction of the separate condenser) that allowed for a drastic reduction in coal consumption.

The Newcomen engine, as improved by John Smeaton in the early 1770s, was capable of a duty between 7 and 10 millions (lbs.). Watt initially raised the duty to 18 millions and later, when his new engine design was fully refined, to 26 millions. Such an economy of fuel made profitable the use of the steam engine in mines situated in locations where coal was expensive. In fact, the first important market for the engine developed by Watt was the Cornish copper and tin mining district. In Cornwall, coal had to be imported from Wales by sea and was extremely expensive. Between 1777 and 1801, Boulton and Watt erected 49 pumping engines in the mines of Cornwall. Jennifer Tann has described the crucial role of the “Cornish business” for the fortunes of the two partners in these terms:

Whether the criterion is the number of engines, their size or the contribution to new capital, Cornish engines comprised a large proportion of Boulton & Watt’s business during the late 1770s to mid 1780s. From 1777 to 1782, Cornish engines accounted for more than 40% of Boulton & Watt’s total business and in some years the figure was significantly higher. In the early 1780s Cornish business was more fluctuating but with the exception of 1784, Cornish engines accounted for between 28% and 80% of Boulton & Watt’s business (Tann, 1996, pp. 29-30).

The typical agreement that Boulton & Watt stipulated with the Cornish mine entrepreneurs (commonly termed “adventurers”) was that the two partners would provide the drawings and supervise the works of erection of the engine. They would also supply some particularly important components of the engine (such as some of the valves). These expenditures would have been charged to the mine adventurer at their cost (i.e. not including any profit for Boulton & Watt). In addition, the mine adventurer had to buy the other components of the engine not directly supplied by the two partners and to build the engine house. These were all elements of the total fixed cost associated with the erection of a Boulton & Watt engine.

The profits for Boulton & Watt resulted from the royalties they charged for the use of their engine. Watt’s invention was protected by the patent for the separate condenser he took out in 1769, which an Act of Parliament prolonged until 1800. The pricing policy of the two partners was to charge an annual premium equal to one-third of the savings of the fuel costs attained by the Watt engine in comparison to the Newcomen engine. This required a number of quite

complicated calculations, aimed at identifying the *hypothetical* coal consumption of a Newcomen engine supplying the same power of that of the Watt engine installed in the mine.

At the beginning, this type of agreement was accepted on very favourable terms by the mine adventurers. However, after some time, the pricing policy of Boulton and Watt was perceived as extremely oppressive. Firstly, the winter months during which most water had to be pumped out (and, consequently, the highest premiums had to be paid) were the ones in which mines were least productive. Secondly, mine adventurers knew the amount of payments they owed to Boulton and Watt only after these had matured. Finally, in the late eighteenth century, several engineers in Cornwall had begun to work on further improvements to the steam engine, but their attempts were frustrated by Boulton and Watt's absolute refusal to license their invention. The most famous case in this respect was that of Jonathan Hornblower who had erected the first compound engine in 1781 and who found the further development of his invention obstructed by the actions of Boulton and Watt (Jenkins, 1931; Torrens, 1994).²⁶

Watt's patent was very broad in scope (covering all engines making use of the separate condenser *and* all engines using steam as a "working substance"). In other words, the patent had a very large blocking power. The enforcement of almost absolute control on the evolution of steam technology, using the wide scope of the patent, became a crucial component of Boulton and Watt's business strategy. This strategy was motivated by the peculiar position of the company (as consulting engineers decentralizing the major part of engine production). All in all, it seems quite clear that Watt's patent had a highly detrimental impact on the rate of innovation in steam technology (Kanefsky, 1978).

²⁶ Boulton and Watt tried, through their agent in Cornwall Thomas Wilson, to discredit the performance of the Hornblower engine in advertisements published in local newspapers and in two pamphlets addressed to the mine owners. Note that this might be considered as a typical example of the Fud (fear, uncertainty and doubt) strategy so much chastised by Microsoft opponents. Boulton & Watt also applied successfully to the Parliament for the rejection of Hornblower's request of extension of his patent. See Jenkins (1931).

After having considered the idea of submitting a petition to Parliament asking for the repeal of the Act that prolonged the duration of Watt's patent, in the 1790s, Cornish adventurers decided to explicitly challenge its validity by installing a number of "pirate" engines erected by local engineers.⁴ A lengthy legal dispute followed. The dispute ended in 1799 with the courts confirming the legal validity of Watt's patent and, in this way, attributing a complete victory to Boulton & Watt. The dispute also had other far-reaching consequences. Boulton & Watt, with their legal victory (pursued with relentless determination), completely alienated any residual sympathy towards them in Cornwall. After the expiration of Watt's patent in 1800, steam engine orders to Boulton & Watt from Cornish mines ceased completely and the two partners had to call their agent in the county back to Birmingham.

Following the departure of Boulton and Watt, the maintenance and the improvement of Cornish pumping engines underwent a period of "slackness", as the mine adventurers were content with the financial relief coming from the cessation of the premia. This situation lasted until 1811, when a group of mine "captains" (mine managers) decided to begin the publication of a monthly journal reporting the salient technical characteristics, the operating procedures and the performance of each engine. The explicit intention was twofold. First the publication would permit the rapid individuation and diffusion of best-practice techniques. Secondly, it would create a climate of competition among the engineers entrusted with the different pumping engines, with favourable effects on the rate of technical progress.

Joel Lean, a highly respected mine captain, was appointed as the first "engine reporter". The publication was called *Lean's Engine Reporter*. After his death, the publication of the reports was continued by his son and lasted until 1904.

As Cardwell has aptly noticed:

The publication of the monthly *Engine Reporter* seems to have been quite unprecedented, and in striking contrast to the furtive secrecy that had surrounded so many of the notable improvements to the steam engine. It was a co-operative endeavour to raise the standards of all engines everywhere by publishing the details of the performance of each one, so that that everybody could see which models were performing best and how much. ²⁷

Thus, the very publication of *Lean's Reporter* seems indeed to mark the transition from a proprietary technical knowledge regime to a new collective invention one.

Concomitant with the beginning of the publication of *Lean's Engine Reporter*, Richard Trevithick and Arthur Woolf began erecting high-pressure engines in Cornish mines. The layout of the engine designed in 1812 by Richard Trevithick at the Wheal Prosper mine soon became the basic one for Cornish pumping engines. Interestingly enough, Trevithick did not patent his high pressure engine:

Trevithick only regarded this engine as a small model designed to demonstrate what high-pressure could do. He claimed no patent rights for it; others were free to copy it if they would (Rowe, 1953, p.124).

Following the publication of the engine reports, the thermodynamic efficiency of Cornish engines improved steadily. On strictly engineering grounds, this amounted to a very effective exploration of the merits of the use of high-pressure steam centred on the designs originally introduced by Trevithick and Woolf . Figure 1 displays the evolution over time of the efficiency of Cornish steam engines (based on the collation of several sources).

Figure 1 around here

The figure clearly indicates that the practice of information sharing resulted in a marked acceleration in the rate of technical advance. As in the case of the Cleveland iron industry described by Allen, the rate of innovation in Cornish engines appears to be tightly linked with the rate of capital formation. Installation of new productive capacity permitted experimentation with design alterations facilitating the discovery of new improvements. Hence, the period of high duty

²⁷ Cardwell (1971), p.156.

growth coincided with the rapid expansion of the Cornish mining industry, conversely the phase of recession after the 1850s translated itself into a slow decline of the average duty (Barton, 1968 and 1969).

It is worth noticing that, although centred around the basic layout ideated by Trevithick, the design of the Cornish pumping engines remained always in a sort of fluid state. With the term “fluid”, we mean that Cornish engineers, after Trevithick, did not stop exploring design modifications. This facilitated a more thorough exploration of the space of technological opportunities, avoiding the risk of remaining trapped in a local optimum configuration.²⁸

Interestingly enough, in the contemporary engineering literature, engines built on the basis of this design layout were not ascribed to this or that particular engineer, but simply known as “Cornish” engines, correctly acknowledging the cooperative and cumulative character of this particular form of technological development.

According to Raymond, this continuous process of localized exploration of the design space is also one of the main advantages of open source software development:

It is not only debugging that is parallelizable; development and (to a perhaps surprising extent) exploration of the design space, too. When your development mode is rapidly iterative, development and enhancement may become special cases of debugging – fixing “bugs of omission” in the original capabilities or concept of the software. Even at a higher level of design, it can be very valuable to have the thinking of many co-developers random walking through the design space near your product.²⁹

The available evidence also suggests that the three conditions that Allen considers as crucial prerequisites for the emergence of a sustainable collective invention were amply satisfied in the Cornish mining district.

Firstly, analogously with the blast furnace case, the design of steam engine was a rather risky undertaking from an engineering point of view. Technology was much ahead than scientific

²⁸ On the technological history of the Cornish pumping engine see Barton (1965). One might also note that this process of continuous design modification makes difficult, for technology historians, to devise a clear-cut technological definition of the Cornish steam engine.

²⁹ Raymond (1999), p. 51.

understanding and the overall performance of a pumping engine could be affected by a host of factors (boiler, steam pressure, engine, pitwork, etc.). Engineers could not rely on a solid theoretical principles when they had to design a new steam engine. The best they could was to extrapolate from the relative performance of existing designs. In such a case, the release of information greatly improved the exploration of the space of technological opportunities.

Secondly, as in the case of Cleveland blast furnaces, Cornish pumping engines were commissioned on a one-off basis to independent consulting engineers. In this way, the very publication of the *Engine Reporter* acted indeed as a “credit file” that steam engineers could use to signal their talents.

Finally, mine owners, had often shares of different mines and, for this reason, were more interested in improvements of the aggregate profitability of the district (the development of a local technology such as the Cornish engine was one way of achieving this). Furthermore, as in Cleveland, improvement in the *average aggregate* performance of the Cornish engines installed, had the positive side effect of increasing the value of the Cornish ore deposits.

To sum up, the peculiar organisation of the Cornish mining industry made mine entrepreneurs interested in improvements of the *aggregate average performance* of the pumping engines used and, at the same time, engineers in publicly signalling the *above average performance* of the engines they had erected. Thus, *Lean's Engine Reporter* successfully reconciled the tensions between collaboration (among mine adventurers) and competition (among engineers) operating in the Cornish mining district.

Besides these factors, it is quite clear that the transition to a collective invention regime in Cornwall was also motivated by the disappointing experience of the Boulton & Watt monopoly

period. After the beginning of the publication of *Lean's Engine Reporter*, Cornish engineers followed the example of Trevithick with his Wheal Prosper engine and normally preferred not to take out patents for their inventions. Table 1 reports the geographical distribution (measured using the stated addresses of the patentees) of patents in steam power technology over the period 1698-1852 (see Andrew et al. 2001 for a detailed quantitative analysis of the pattern of steam power patenting over the entire nineteenth century).

Table 1 about here

The London and Middlesex area holds the predominant position. In this respect the pattern of patenting in steam technology mirrors that for overall patenting outlined by Christine MacLeod (1988, pp.119-124), and it is likely that this high number is mainly explained both by the growth of the metropolis as a commercial and manufacturing centre and by the proximity to the patent office, which gave would-be patentees the possibility of following closely the administrative procedures related to the granting of the patent. Surrey also has a quite high concentration of steam patents. This case, besides by the proximity to the patent office, may also be accounted for by the presence in the area of a number of engineering firms specialized in the production of capital goods (MacLeod, 1988, p. 124). Other notable locations with high numbers of steam patents are Warwickshire, Lancashire and Yorkshire, where patents were probably related to the increasing use of steam power by the industries there located. Again, one should take into account that in this case as well, patents were essentially an urban phenomenon (MacLeod, 1988, p. 125) and so they were concentrated in major towns such as Birmingham, Liverpool, Manchester and Leeds. The table also reports the number of patents in major urban centres.

Over the entire period 1698-1852, the share of Cornwall in total patenting is 1.85 per cent, which does not reflect at all the major contribution of the county to the development of steam power

technology. Breaking down the period 1698-1852 into two sub-periods (1698-1812 and 1813-1852), in order to take into account the publication of *Lean's Engine Reporter* is even more revealing. In the first period, Cornwall (including in the count also the patents taken out by Arthur Woolf who, at the time, was working for the Meux & Reid brewery in London) is the county with highest number of patents after the London and Middlesex area, with a share of 9.38 per cent. In the second period, the share of Cornwall drops to a negligible 0.89 per cent and this is exactly the period during which the Cornish pumping engine was actually developed. In our view, this finding is indicative of the widely perceived awareness in the county of the benefits stemming from the adoption of a collective invention regime for the rate of innovation. After the unfortunate experience with the Boulton and Watt monopoly, it seems quite clear that in the Cornish engineering community, an *ethos* prescribing the full release of technical innovations into the public domain emerged and became progressively established.

The case of Arthur Woolf is particularly illustrative. Woolf was one of the leading figures in the Cornish engineering community (Harris, 1966). Born in Cornwall, he had an initial apprenticeship with steam engineering by working with Jonathan Hornblower. In the first decade of nineteenth century he moved to London, where he was entrusted with the steam engines of the Meux & Reid brewery. In this period Woolf took out four patents for innovations in steam engines (in particular his famous compound engine patented in 1804). In 1812 he moved back to Cornwall, where he tried to commercialise his compound engine by means of an agreement similar to the one proposed by Boulton & Watt (royalties paid as a proportion of fuel savings). His initiative was unsuccessful. Most mine adventurers awaited the expiration of the patent in 1818 before installing this type of engine (Farey, 1971, pp.188-189). Later on, in 1823, Woolf invented a new valve for steam engines (the double-beat valve). The adoption of this type of valve greatly facilitated the operation of the engine (Hills, 1989, pp. 109-110). He did not claim any patent right for this invention.

Another example that confirms the negative attitude towards patents existing in the Cornish mining district is the limited diffusion of the two-cylinder compound engine patented by the Cornish engineer, James Sims, in 1841. The first engine of this type erected at the Carn Brea mine performed particularly well in terms of duty (it was the second best engine in the *Reporter* in the early 1840s). However, being a patented design made the engine quite unpopular with other engineers and mine-owners, who, in the end, preferred not to adopt it (Barton, 1969, pp. 110-112).

Passages in the contemporary engineering literature also indicate this consciousness. For example, John Taylor (a leading mine entrepreneur) wrote in 1830:

Under such a system [the *Lean's Engine Reporter*] there is every kind of proof that the application of steam has been improved, so as to greatly economise fuel in Cornwall, and also the rate of improvement has been fairly expressed in the printed reports.....[A]s since the time of Boulton and Watt, no one who has improved our engines has reaped pecuniary reward, it is at least fair, that they should have credit of their skill and exertion. We [adventurers] are not the partisans of any individual engineer or engine maker; we avail ourselves of the assistance of many; and the great scale upon which we have to experiment makes the result most interesting to us. (Taylor quoted in Farey, 1971, pp. 251-252).

5. Concluding remarks

In general terms, current conventional wisdom considers strong and broad intellectual property rights as a key-stimulus to technical progress. Strong intellectual property rights are deemed to constitute an indispensable incentive for motivating an adequate level of private investment in the search for new technologies. At the same time, broad (and well specified) intellectual property rights promote relatively ordered exploration of the space of technological opportunities.³⁰

The evidence presented in this paper seems however to cast doubts on the general validity of such a proposition. In fact, in industries (such as the cases examined in this paper) where the dynamics of technological change displays a cumulative and incremental character, the protection of “commons” of freely accessible knowledge is likely to yield to much higher rates of

³⁰ See Mazzoleni and Nelson (1998) for a (rather sceptical) overview of the theoretical arguments and the empirical evidence supporting such a proposition.

innovation, than the enforcement of strong intellectual property rights.³¹ In such instances, even at the level of the individual actor, free sharing of technological knowledge may be a much more rewarding strategy than secrecy or individual appropriation and commercialization.

This can account for the emergence of particular institutional set-ups (which following Allen might be termed “collective invention regimes”) ensuring that new technological knowledge remains in the public domain. An additional feature of these institutional arrangements is the creation of a system for the public acknowledgement of the merits of the various individual contributions.

In this paper, we have merely assembled some suggestive evidence and put forward some interpretative hypotheses. Thus, the cases presented here are just indicating what seems to us a very promising research agenda. Hopefully, the recent upsurge of interest in open source software can provide the stimulus for further research in this direction, leading us towards the achievement of a deeper understanding of the emergence and consolidation of collective invention regimes.³²

³¹ Merges and Nelson (1994) have provided a number of examples, showing that in what they call “cumulative systems technologies (i.e. technologies are constituted by systems composed by many interacting elements and where innovations are, most of the times, refinements and improvements of previous inventive steps) the enforcement of a strong regime of intellectual property rights stifled technical progress. In some cases, the deadlock was overcome only when the Government stepped in, inducing the main players in the industry to stipulate agreements prescribing automatic or semi-automatic cross licences of patents. Heller and Eisenberg (1998) considers the case of biomedical research in a similar perspective.

³² See Von Hippel (2001) for similar considerations.

References

- Allen, R. C. (1983), "Collective Invention", *Journal of Economic Behavior and Organization*, **4**, pp.1-24.
- Barton, D. B. (1968), *A History of Copper Mining in Cornwall and Devon*, D. B. Barton, Truro.
- Barton, D. B. (1969), *The Cornish Beam Engine*, D. B. Barton, Truro.
- Cardwell, D. S. L (1971), *From Watt to Clausius*, Heinemann, London.
- Dasgupta, P. and P. A. David (1994), "Toward a New Economics of Science", *Research Policy*, **23**, pp. 487-521.
- Dickinson, H and R. Jenkins (1927), *James Watt and the Steam Engine*, Clarendon, Oxford.
- Farey, J. (1971), *Treatise on the Steam Engine* Vol. II., David & Charles, Newton Abbot.
- Foray, D. (2000), "'Open Knowledge Institutions' and the Economics of Knowledge Transfer", mimeo.
- Harhoff, D., Henkel, J. and E. Von Hippel (2002), "Profiting from Voluntary Information Spillovers: How Users Benefit by Freely Revealing Their Innovations", mimeo.
- Harris, T.R. (1966), *Arthur Woolf, the Cornish Engineer, 1766-1837*, D. B. Barton, Truro.
- Heller, M. A. and R. S. Eisenberg (1998), "Can Patents Deter Innovation? The Anticommons in Biomedical Research", *Nature*, **280**, pp. 698-701.
- Hills, R. L. (1989), *Power from Steam*, Cambridge University Press, Cambridge.
- Jenkins, R. (1931), "Jonathan Hornblower and the Compound Engine", *Transactions of the Newcomen Society*, **11**, 1930-31, pp.138-155.
- Kanefsky, J (1978) "Boulton and Watt and the Development of the Steam Engine: a Reassessment", Exeter Research Group, Discussion Paper 01.
- Lerner, J. and J. Tirole (2002), "Some Simple Economics of Open Source", *Journal of Industrial Economics*, **50**, pp.197-234.
- Lerner, J. and J. Tirole, J. (2001), "The Open Source Movement: Key Research Questions", *European Economic Review*, **45**, pp.819-826.
- Lundvall, B. A. (1988), "Innovation as an Interactive Process: from User-Producer Interaction to the National System of Innovation" in G.Dosi, C. Freeman, R. Nelson, G. Silverberg. L. Soete (eds.), *Technical Change and Economic Theory*, Pinter, London.
- MacLeod, C. (1988), *Inventing the Industrial Revolution*, Cambridge University Press, Cambridge.
- Mazzoleni R. and R.R. Nelson (1998), "The Benefits and Costs of Strong Patent Protection: a Contribution to the Current Debate", *Research Policy*, **27**, pp.273-284.

- Merges, R. P. (1996), "A Comparative Look: at Intellectual Property Rights and the Software Industry", in D.C. Mowery (ed.), *The International Computer Software Industry*, Oxford University Press, Oxford.
- Merges, R.P. and R. R. Nelson(1994), "On Limiting or Encouraging Rivalry in Technical Progress: the Effect of Patent-Scope Decisions", *Journal of Economic Behaviour and Organization*, **25**, pp. 1-24.
- Moody, G. (2001), *Rebel Code. Linux and the Open Source Revolution*. Allen Lane: London.
- Nelson R.R. and Winter S. G. (1977), "In Search of a Useful Theory of Innovation", *Research Policy*, **6**, pp. 36-76.
- Nuvolari, A. (2002), "Collective Invention during the British Industrial Revolution: the Case of the Cornish Pumping Engine", *Cambridge Journal of Economics*, (forthcoming).
- Pole, W. (1844), *A Treatise on the Cornish Pumping Engine*, (London, 1844)
- Raymond, E. S.(1999), *The Cathedral and the Bazaar*, O'Reilly: Sebastopol, California.
- Rosenberg, N. (1982), *Inside the Black Box*, Cambridge: Cambridge University Press.
- Rowe, J. (1953), *Cornwall in the Age of the Industrial Revolution*, Liverpool: Liverpool University Press.
- Salus, P. H. (1994), *A Quarter Century of Unix*, Addison-Wesley: Reading, Massachusetts.
- Stallman, R. M. (1999), "The GNU Operating System and the Free Software Movement", in DiBona C., Ockman S. and Stone M. (eds.), *Open Sources: Voices from the Open Source Revolution*, O'Reilly: Sebastopol, California.
- Steinmueller, W. E. (1996), "The US Software Industry: an Analysis and Interpretive History" in D.C. Mowery (ed.), *The International Computer Software Industry*, Oxford University Press: Oxford.
- Tann, J. (1996), "Riches from Copper: the Adoption of the Boulton and Watt Engine by Cornish Mine Adventurers", *Transactions of the Newcomen Society*, Vol.67 (1995-6), pp.27-51.
- Torrens, H. (1994), "Jonathan Hornblower and the Steam Engine: a Historiographic Analysis", in D. Smith (ed.), *Perceptions of Great Engineers: Fact and Fantasy*, Science Museum, London.
- Torrise, S. (1998), *Industrial Organisation and Innovation. An International Study of the Software Industry*, Edward Elgar: Cheltenham.
- Torvalds, L. B. (1999), "Interview with Linus Torvalds: What Motivates Free Software Developers ?", http://www.firstmonday.dk/issues/issue3_3/torvalds/index.html (accessed march 26, 2001).
- Valloppillil, V. (1998), "Open Source Software: a (New ?) Development Methodology" [known as the "Halloween Document"], <http://www.opensource.org/halloween> (accessed march 31, 2001).
- Von Hippel, E. (1987), "Cooperation between Rivals: Informal Know-How Trading", *Research Policy*, **16**, pp.291-302.

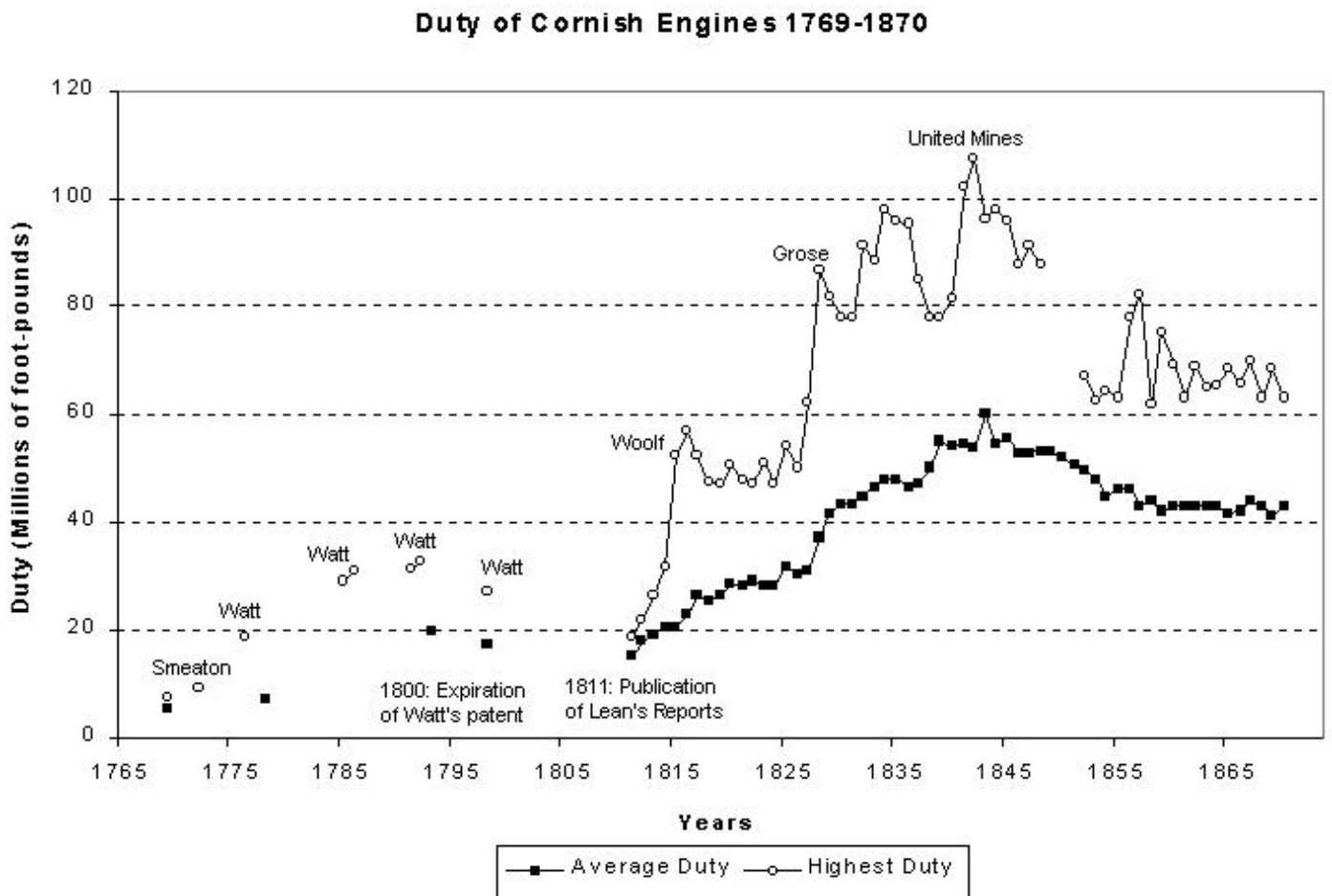
Von Hippel, E. (2001), "Innovation by Users Communities: Learning from Open Source Software", *MIT Sloan Management Review*, **42** (4), pp. 82-86.

Von Tunzelmann, G. N. (1970), "Technological Diffusion during the Industrial Revolution: the Case of the Cornish Pumping Engine" in R. M. Hartwell (ed.), *The Industrial Revolution*, Basil Blackwell, Oxford.

Von Tunzelmann, G. N. (1978), *Steam Power and British Industrialization to 1860*, Clarendon, Oxford.

Young, R. (1999), "Giving It Away: How Red Hat Stumbled Across a New Economic Model and Helped Improve an Industry" in DiBona C., Ockman S. and Stone M. (eds.), *Open Sources: Voices from the Open Source Revolution*, O'Reilly: Sebastopol, California.

Figure 1: Duty of Cornish Engines



Sources: Lean (1839), Pole (1844), Dickinson and Jenkins (1927), Barton (1969)

Table 1: Geographical Distribution of British Steam Engine Patents, 1698-1852

County/Location	N.of Patents		%		N.of Patents		%	
	1698-1852	1698-1852	1698-1812	1698-1812	1813-1852	1813-1852	1813-1852	1813-1852
Cheshire	14	1.23	0	0	14	1.39		
Cornwall	17	1.50	8	6.25	9	0.89		
Cornwall*	21	1.85	12	9.38	9	0.89		
Derby	11	0.97	1	0.78	10	0.99		
Durham	13	1.15	0	0	13	1.29		
Essex	6	0.53	0	0	6	0.60		
France	21	1.85	0	0	21	2.09		
Gloucester	20	1.76	8	6.25	12	1.19		
-Bristol	12	1.06	4	3.13	8	0.79		
Hampshire	9	0.79	0	0	9	0.89		
Ireland	13	1.15	1	0.78	12	1.19		
Kent	31	2.73	1	0.78	30	2.98		
Lancashire	145	12.78	5	3.91	140	13.90		
-Liverpool	35	3.08	1	0.78	34	3.38		
-Manchester	58	5.11	2	1.56	56	5.56		
London & Middlesex	395	34.80	40	31.25	355	35.25		
Northumberland	22	1.94	2	1.56	20	1.99		
-Newcastle-up.-Tyne	11	0.97	1	0.78	10	0.99		
Nottingham	13	1.15	1	0.78	12	1.19		
Scotland	47	4.14	6	4.69	41	4.07		
-Edinburgh	9	0.79	0	0	9	0.89		
-Glasgow	22	1.94	3	2.34	19	1.89		
Shropshire	6	0.53	3	2.34	3	0.30		
Somerset	4	0.35	2	1.56	2	0.20		
-Bath	2	0.18	1	0.78	1	0.10		
Stafford	27	2.38	5	3.91	22	2.18		
Suffolk	5	0.44	0	0	5	0.50		
Surrey	88	7.75	10	7.81	78	7.75		
USA	13	1.15	2	1.56	11	1.09		
Wales	12	1.06	1	0.78	11	1.09		
Warwick	58	5.11	8	6.25	50	4.97		
-Birmingham	55	4.85	6	4.69	49	4.87		
Worcester	11	0.97	1	0.78	10	0.99		
York	63	5.55	11	8.59	52	5.16		
-Bradford	11	0.97	0	0	11	1.09		
-Kingston-up.-Hull	9	0.79	2	1.56	7	0.70		
-Leeds	17	1.50	3	2.34	14	1.39		
-Sheffield	6	0.53	0	0	6	0.60		
Others	71	6.26	12	9.38	59	5.86		
Total	1135	100	128	100	1007	100		

* Cornwall including the patents taken by Arthur Woolf.

Source: The list of steam engine patents is taken from *Abridgments of Specification relative to the Steam Engine*, London, 1871. In order to retrieve the stated residence of the patentees, these patents have been matched with those contained in B. Woodcroft, *Titles of Patents of Invention Chronologically Arranged*, London, 1854.



Eindhoven Centre for Innovation Studies

WORKING PAPERS

Ecis working papers 2002-2003 (January 2003):

- 02.01 M. van Dijk
The Determinants of Export Performance in Developing countries: The Case of Indonesian manufacturing
- 02.02 M. Caniels & H. Romijn
Firm-level knowledge accumulation and regional dynamics
- 02.03 F. van Echtelt & F. Wynstra
Managing Supplier Integration into Product Development: A Literature Review and Conceptual Model
- 02.04 H. Romijn & J. Brenters
A sub-sector approach to cost-benefit analysis: Small-scale sisal processing in Tanzania
- 02.05 K. Heimeriks
Alliance Capability, Collaboration Quality, and Alliance Performance: An Integrated Framework.
- 02.06 G. Duysters, J. Hagedoorn & C. Lemmens
The Effect of Alliance Block Membership on Innovative Performance
- 02.07 G. Duysters & C. Lemmens
Cohesive subgroup formation: Enabling and constraining effects of social capital in strategic technology alliance networks
- 02.08 G. Duysters & K. Heimeriks
The influence of alliance capabilities on alliance performance: an empirical investigation.
- 02.09 J. Ulijn, D. Vogel & T. Bemelmans
ICT Study implications for human interaction and culture: Intro to a special issue
- 02.10 A. van Luxemburg, J. Ulijn & N. Amare
The Contribution of Electronic Communication Media to the Design Process: Communicative and Cultural Implications
- 02.11 B. Verspagen & W. Schoenmakers
The Spatial Dimension of Patenting by Multinational Firms in Europe
- 02.12 G. Silverberg & B. Verspagen
A Percolation Model of Innovation in Complex Technology Spaces

- 02.13 B. Verspagen
Structural Change and Technology. A Long View
- 02.14 A. Cappelen, F. Castellacci, J. Fagerberg and B. Verspagen
The Impact of Regional Support on Growth and Convergence in the European Union
- 02.15 K. Frenken & A. Nuvolari
Entropy Statistics as a Framework to Analyse Technological Evolution
- 02.16 J. Ulijn & A. Fayolle
Towards cooperation between European start ups: The position of the French, Dutch, and German entrepreneurial and innovative engineer
- 02.17 B. Sadowski & C. van Beers
The Innovation Performance of Foreign Affiliates: Evidence from Dutch Manufacturing Firms
- 02.18 J. Ulijn, A. Lincke & F. Wynstra
The effect of Dutch and German cultures on negotiation strategy comparing operations and innovation management in the supply chain
- 02.19 A. Lim
Standards Setting Processes in ICT: The Negotiations Approach
- 02.20 Paola Criscuolo, Rajneesh Narula & Bart Verspagen
The relative importance of home and host innovation systems in the internationalisation of MNE R&D: a patent citation analysis
- 02.21 Francis K. Yamfwa, Adam Szirmai and Chibwe Lwamba
Zambian Manufacturing Performance in Comparative Perspective
- 03.01 A. Nuvolari
Open source software development: some historical perspectives