CENTER FOR ECONOMIC AND POLICY RESEARCH

Opening Doors and Smashing Windows: Alternative Measures for Funding Software Development

By:

Dean Baker

October 2005

Center for Economic and Policy Research
1611 Connecticut Avenue, NW
Suite 400
Washington, D.C. 20009
Tel: 202-293-5380
Fax: 202-588-1356
www.cepr.net

Contents

About the Authors

Dean Baker is Co-Director of the Center for Economic and Policy Research.

Acknowledgments

John Schmitt provided helpful comments on an earlier version of this paper.

Executive Summary

Copyrights and patents are forms of government intervention in the market that are relics of the medieval guild system. They are an outdated and inefficient means to support creative and innovative work in the 21st century. These government-granted monopolies lead protected software to sell at prices that are far above the free-market price. In most cases, in the absence of copyright and patent protection, software would be available over the Internet at zero cost.

This paper examines the ways in which copyrights and patents (intellectual property rights, or IPRs) lead to inefficiency and waste, and outlines alternative mechanisms for supporting software development. It shows that:

- Copyright and patent protection may impose costs on consumers of between $80 and $120 billion a year, compared to a situation in which all software was available at its competitive-market price. The pure efficiency loss to the economy could be in the neighborhood of $70 to $110 billion a year. These losses dwarf estimates of the losses from other forms of protectionism, such as tariffs or quotas on imported goods;
- A substantial portion of the resources devoted to software development are currently wasted due to IPRs. This is a result of the fact that IPR protection leads to unnecessary duplication, as developers have substantial incentive to produce software that simply replicates the function of existing software. In the absence of IPR protection, developers could better spend their time improving existing software. IPRs also provide incentives for software locks and secrecy, which impede the process of software development. It is likely that a substantial portion of software development (possibly a majority) is misdirected as a result of the market distortions created by IPRs;
- IPRs also lead to large amounts of waste by providing incentives for rent-seeking activity. This waste includes expenditures for advertising and marketing, and payments to lawyers and lobbyists. Those enjoying IPR protection have also been able to impose costs on third parties, for example by requiring Internet Service Providers to monitor activities or universities to take steps to reduce the amount of unauthorized copying of IPR protected material on their premises. IPR holders have also secured laws that restrict the development of software and hardware designed to support better searches and digital reproductions;
- There are feasible alternative mechanisms for supporting software development. One mechanism outlined in the paper would create a "Software Development Corps," which would be a series of competing government funded software corporations. An annual appropriation of $2.1 billion (approximately 0.08 percent of federal spending) should be enough to support the work of approximately 20,000 software developers. The government should be able to recoup most, if not all, of this money through the lower price it will pay on the computers and software it purchases. The remaining benefit would be the equivalent of a tax cut to consumers in the range of $80 to $120 billion a year. This money would provide a substantial stimulus to the economy and lead to the creation of millions of jobs.

A unique feature of this proposal for a Software Development Corps is that it could exist side-by-side with the existing copyright and patent system. Software developers, such as Microsoft, could continue to produce IPR-protected software and sell it in the market. If they actually produce software that is sufficiently superior to justify paying the IPR- protected price, then consumers will still purchase it. In other words, the alternative mechanisms described in this paper provide a basis for a market test of the relative efficiency of IPR-supported research. Those who believe that IPRs are the most efficient mechanism for supporting software development should favor this sort of test, which would prove their case.

Introduction

The computer software industry is usually considered to be at the cutting edge of U.S. technology. At the same time, the industry relies largely on government protectionism in the form of copyrights and patents, relics of the feudal guild system, to finance the development of new software. The reliance on these antiquated mechanisms leads to both economic inefficiency for the economy as a whole, and poorer quality software than would be possible in an environment in which all software was placed in the public domain.

This paper outlines the ways in which patents and copyrights in software lead to economic inefficiency. It also presents alternative mechanisms that could be used to finance the software development that is currently supported by copyright or patent protection. Specifically, it suggests that a mix of a system of direct government funding for software development and a system of individual vouchers could be a more efficient mechanism for financing the development of new software. All the new software developed under both systems would be placed in the public domain so that it could be used at zero cost and freely modified by other developers.[1]

[1] There is an important issue about how best to keep software developed with public funds in the public domain in a world where private individuals can still get IPRs. One possible mechanism is the "copyleft" system developed by the Free Software Movement. Under the copyleft system, the software is copyrighted, but can still be freely reproduced, as long as an subsequent refinements of the software are made freely available either by being placed in the public domain or being subject to a new copyleft. A fuller description can be found at http://www.gnu.org/copyleft/gpl.html

The Inefficiency of Copyright and Patent Protection in the Software Industry

One of the most basic principles in economics is that efficiency is maximized when products sell at their marginal cost of production. Copyright and patent protection is explicitly designed to prevent marginal-cost pricing by providing copyright and patent holders with a government-enforced monopoly on the protected product. This monopoly allows the holder of these intellectual property rights (IPR) to charge prices that are far above the marginal cost of production, which would be the expected equilibrium price in a freely competitive market.

Of course, this inefficiency may make economic sense if the monopoly provides incentives to innovate. However, the overriding policy question is whether copyright and patent monopolies are the *best way* to provide incentives for software development. To make this assessment, it is necessary to determine the costs of IPRs in software and compare them to the cost of other mechanisms for financing software development.

There are three distinct ways in which IPRs in software lead to economic inefficiency:

1) The gap between the IPR-protected price and the competitive-market price (which would be zero for most software, since it can be transferred costlessly over the Internet) leads to a deadweight efficiency loss. There are many consumers who would be willing to pay at least the marginal cost of producing another copy of the software, but not the IPR-protected price. [This is exactly the same argument that is made for the benefit of free trade, although the inefficiencies that result from most tariffs, which are in the range of 10-20 percent, are trivial compared to the inefficiencies that result from IPR protection.]

2) The fact that certain types of software are proprietary, and cannot be freely adapted to develop new and better software, impedes the process of software development. Research proceeds most rapidly when all findings are publicly available and can be freely shared by the community of researchers. The fact that some software is proprietary also leads to duplication of effort, which would serve no purpose, if all software were freely available.

3) Holders of IPRs in software devote resources to protecting their IPRs, and also force others to devote resources to protect their IPRs. Specifically, software producers spend resources designing locks, contracting services to monitor usage of their software, hiring lobbyists and contributing to politicians to protect their IPRs, hiring lawyers to enforce their IPRs, and demanding that the government and even third parties such as Internet Service Providers and universities take measures to protect their IPRs.

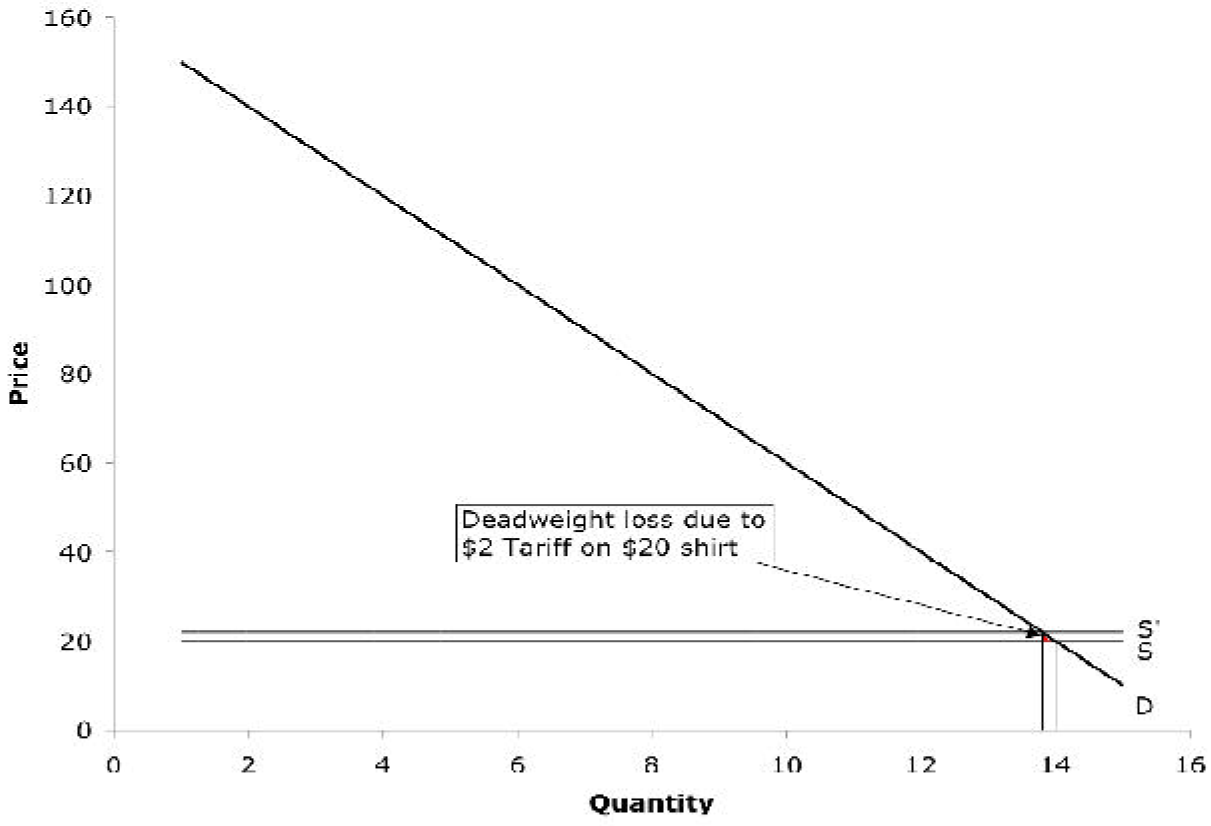
It is worth examining in somewhat more detail the nature and size of the economic costs associated with each of these sources of inefficiency before examining the feasibility of alternative methods of supporting software innovation.

Deadweight Loss: Consumers Cannot Buy Software at the Cost of Production

A vast body of economic research attempts to measure the size of the loss to consumers that results from tariffs or other taxes that create a gap between what it costs sellers to produce a product and the price that consumers have to pay in the market. Economists argue that the gap between the two should be as little as possible, and ideally zero, because this gap is a source of pure waste.

The logic here is simple. Suppose a foreign apparel manufacturer can produce a shirt for $20, counting all costs, including shipping and a normal profit. If there is a 10 percent tariff then the shirt will sell in the United States for $22. This means that consumers who would have been willing to pay more than $20, but less than $22 will not buy the shirt, because of the tariff. This is the loss due to trade protection that leads most economists to oppose tariffs. This loss is a pure loss to the economy – in contrast to the extra $2 paid by people who actually purchase the shirt. In that case, the $2 lost by consumers is transferred to the government. The government can then use this money for some other purpose. However, there is no money transferred when people don't buy the shirt because of the tariff. The people who end up not buying the shirt are simply made worse off because of the tariff.

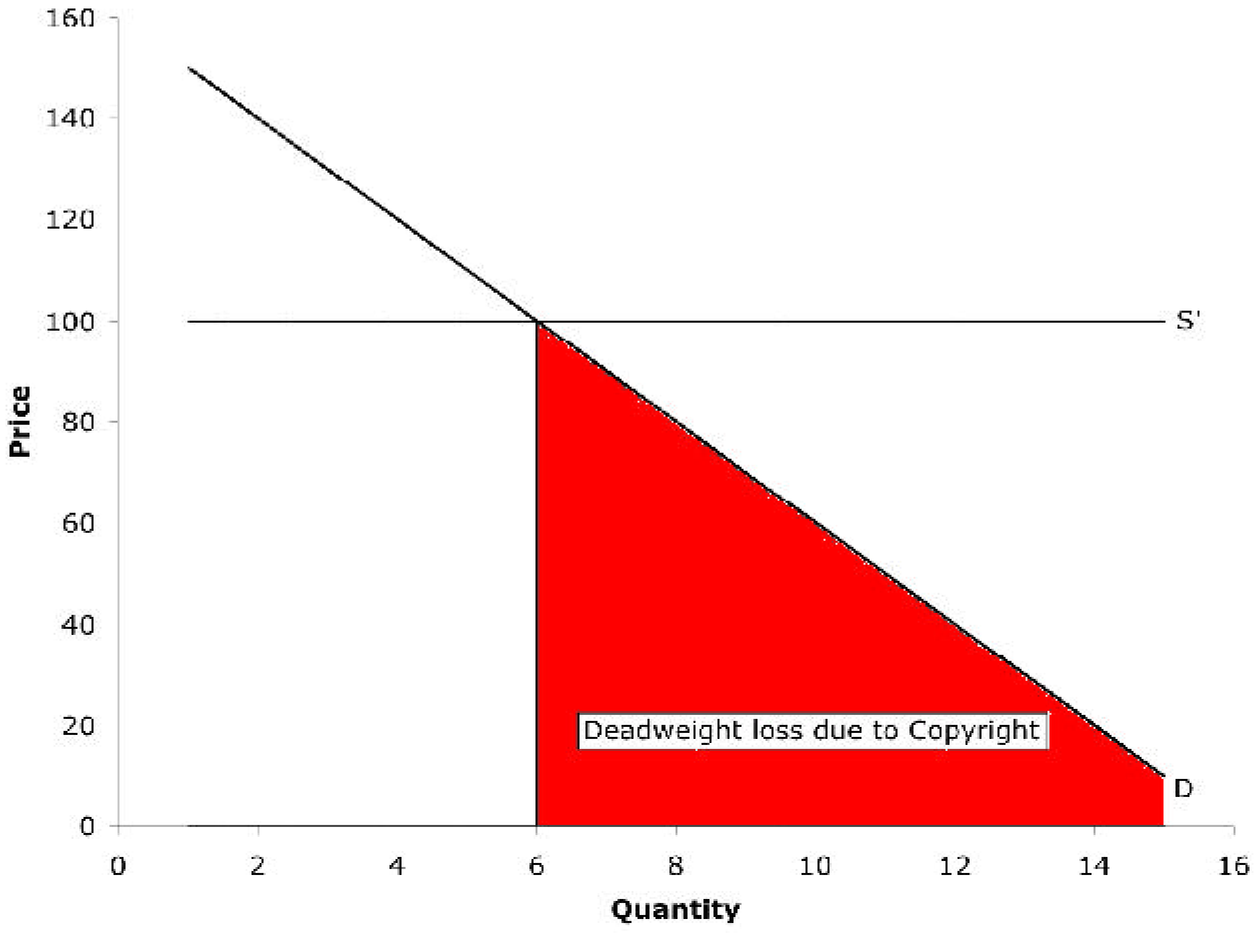
Figure 1
Deadweight Loss From $2 Tariff on Shirts



Identical logic applies to the IPR-protected price of software, except in the case of software, most products would be available at little or no cost in the absence of IPR protection. Instead of raising the price by 10 percent, as might be the case with a typical tariff, with software, the only reason that there is a price at all is because of the IPR protection. Various software applications (e.g. spreadsheets, word processing programs, graphic programs) that would be available for free over the Internet, can often be sold for several hundred dollars, because the government grants a monopoly to the copyright or patent holder. People who try to distribute this software for free face fines, and possibly even criminal charges by the government.

It is possible to use the same basic graph to show the losses from IPR protection with software, but the numbers would be much larger, as shown in the graph below.

Figure 2
Deadweight Loss From Zero Marginal Cost Software, Selling at $100



Since the gap between the cost of production of software and the IPR-protected price is so much larger, the number of consumers who end up not buying IPR protected software is likely to be much larger, and the potential loss to consumers will be much larger as well.

The logic here is straightforward. If a tariff raises the price of a shirt by $2, then the most benefit that any consumer could have lost because of this tariff is just less than $2.00 – say $1.99. If the shirt was in fact worth $2.00 more to the consumer than the competitive market price (i.e. $22) then she would have bought the shirt. There is only an issue if the shirt was worth more than $20 to the consumer, but less than $22 – in which case she would have bought it at the competitive market price, but not at the price charged in the market with a tariff.

In the case of IPR protection of software, there are likely to be many people who might have found the software useful and would have even been willing to pay some price to use it, but less than the $100 IPR-protected price shown in the graph. In this case, the maximum loss to consumers would be just less than the IPR protected price (i.e. up to $100), since that is the most that someone would have been willing to pay for the software, who did not actually buy it.

It is possible to do some rough calculations to get a general estimate of the potential gains to consumers, and efficiency gains to the economy, from eliminating IPR protection in software. Table 1 shows projections of gains from eliminating IPR protection for software in the market for notebook computers, the market for desktop computers, and the market for computer and video games. (For a fuller explanation of these calculations see the appendix.) The projections assume alternatively that demand is relatively unresponsive to price (low elasticity), moderately responsive to price, and highly responsive to price.

Table 1
Gains to Consumers from Eliminating IPR Protection for Computer Software

Middle Elasticity

	Average IPR Price	**Average free market price**	**Increase in Units Sold**	**Gains to Consumers**	**Loss to Producers**	**Net Surplus**
				billion 2005 dollars		
Notebook computers	$1,000	$800*	21.5 million	$34.6	$1.2	$33.4
Desktop computers	$800	$600*	64.4 million	$48.5	$1.6	$46.9
Computer and Video games	$29.50	$0	25.2 billion	$13.3	$7.3	$6.0
Totals				$96.4	$10.1	$86.3

High Elasticity

	Average IPR Price	**Average free market price**	**Increase in Units Sold**	**Gains to Consumers**	**Loss to Producers**	**Net Surplus**
				billion 2005 dollars		
Notebook computers	$1,000	$800*	37.5 million	$37.6	$1.2	$36.4
Desktop computers	$800	$600*	79.0 million	$53.5	$1.6	$51.9
Computer and Video games	$29.50	$0	8.4 trillion	$24.4	$7.3	$17.1
Totals				$115.5	$10.1	$105.4

Source: author's calculations, see appendix.

It is assumed that the price reductions are very large in all cases, which leads to substantial gains to both consumers and to the economy in the form of increased output. The calculations in the table assume that the average sale price of both a notebook and desktop computer would fall by $200 (including installed software applications), if all software were available at zero cost. They also assume that computer purchasers would use an average of $600 of additional software (at current market prices) if all software were available at zero cost. The calculations assume that all video and computer games would be available at zero cost over the Internet.

Table 2 summarizes the gains to consumers and the net efficiency gains (after deducting losses to producers) in the three scenarios.

Table 2
Gains to Consumers and net Gains to the Economy From Eliminating IPRs in Software

	Gains to Consumers	**Net Surplus**
	billion 2005 dollars	
Low Elasticity	$84.2	$74.1
Medium Elasticity	$96.4	$86.3
High Elasticity	$115.5	$105.4

Source: author's calculations, see appendix.

The projected gains in all three cases are extremely large. In the low elasticity case, the gain to consumers is projected as $84.2 billion a year, an amount that is more than 50 percent larger than what the federal government spends on roads each year. The projected gains to consumers in the high elasticity case are $115.5 billion, more than twice the annual federal transportation budget. The projected gains to the economy are similarly large. The net surplus in the low elasticity case is $74.1 billion a year, an amount that exceeds the GDP of Guatemala. The net surplus in the high elasticity case is more than $100 billion a year, an amount that is equal to approximately 0.8 percent of U.S. GDP. Very few economic policies yield these sorts of potential efficiency gains.

There are two important points to note on these calculations, in addition to the fact that they are extremely crude projections. First, they are based on worldwide estimates of the computer market. Much of these projected gains would accrue to people outside of the United States. While there is nothing wrong with this, presumably it would be desirable to devise a method to share the cost of developing software, at least among middle income and wealthy countries, who can afford to pay part of this burden. Devising a system to share the costs of an alternative system for financing software development internationally would require some coordination. However, would likely be far simpler and easier to enforce than the complex system currently in place to standardize and enforce IPRs internationally.

The second point is that these calculations only assume gains associated with the purchase of new computers. If software were freely available over the web, tens of millions of current owners of computers would take advantage of the opportunity to download better software at zero cost. This would lead to a large one-time gain (probably several times larger than the annual gain assumed in these calculations) that is not captured in these projections. This one-time gain in consumer surplus could easily run into the hundreds of billions of dollars, if several hundred million computer owners download software worth an average of several hundred dollars per computer.

IPR Protection – Impeding Innovation and Promoting Duplication

The second major type of inefficiency associated with IPR protections in software are the ways in which IPR protection impedes the development of new software and leads to wasteful duplication. The first point is easy to describe, although difficult to quantify. If we imagine the scientific process as being a collective exercise in which everyone builds on each other's progress, and learns from mistakes, then the process will obviously advance most quickly in an environment in which information flows most freely. If software developers have ready access to the work of other developers, and can freely integrate this work into their own projects, then the development process will proceed much more quickly than if developers are forced to work in isolation or cannot borrow from the discoveries of others.

IPR protection in software leads to both sorts of problems. Companies seeking to profit from software copyrights or patents will not make information on their progress publicly available until they actually file for IPR protection. This is simply the nature of the incentives provided by the IPR system. If a company allowed information to get out before it had filed for IPR protection, then a competitor could use this information to get a copyright or patent of their own. Even when a company does file for IPR protection, it will typically only disclose the information needed to gain protection – it will still keep much of its work secret. Software producers are likely to continue to maintain as much secrecy as possible around their process of software development, both to prevent aiding competitors and to preserve an edge in developing future products.

Of course, even when a product is available, competitors cannot freely build on it if the product is subject to IPR protection,. In other words, a software designer does not have the option to develop an extension of an IPR-protected piece of software, which might make it more useful, without the permission of the IPR holder. This could prevent many useful adaptations of software, since it can be costly and time consuming to negotiate an arrangement with an IPR holder. In many cases, software developers may simply choose to direct their energies elsewhere.

There can also be instances where ambiguity about the nature of a software copyright or patent, by itself, inhibits progress. For example, if Microsoft establishes a reputation for aggressively defending its IPRs, it may discourage software developers from pursuing lines of research that are not actually infringing on Microsoft's IPRs, but which may still be close enough to prompt lawsuits. The asymmetry of resources between Microsoft and a small software developer is sufficiently large that many developers may simply decide to abandon a path of research altogether rather than get into a long legal battle over IPRs.

The existence of IPR protection in software also provides incentives for wasteful duplication. If there is an application subject to IPR protection that proves to be popular (e.g. a word-processing program or spreadsheet program), there is a powerful incentive for other developers to produce a competitive program that does not infringe on the copyright or patent. While there is nothing wrong with giving consumers a choice of programs, in many cases the sole motivation may be that the initial program was subject to IPR protection. In other words, if a popular spreadsheet program were available at zero cost over the web to anyone who wanted to use it, there would be little incentive for someone to attempt to develop an alternative, even if the alternative would be subject to IPR protection. If the original program lacked important features, the most efficient action for a developer to do would be to modify the existing program to incorporate the new feature, rather than create a near-duplicate of the original program —all the while being careful not to infringe on the original developer's IPRs— simply to add an additional feature. However, if the original program is subject to IPR protection, then an alternative can potentially take away a substantial portion of the monopoly profits earned by the first program.

This sort of competition is actually desirable in a world where programs sell for high prices due to IPR protection, since competition will lead to lower prices for consumers. However, in a world without IPR protection, there would be little reason to design a second program that essentially just replicates the tasks already performed by an existing program. In most cases, the time of software designers could be much more productively spent improving the original program, or on other projects.

Unfortunately, it is difficult to estimate the losses due to secrecy in development or unnecessary duplication. In the pharmaceutical industry, approximately two thirds of research spending by the industry goes to develop duplicative rather than breakthrough drugs.[2] While there is some value associated with such copycat drugs (some patients may have bad reactions to one drug, but not to the copycat drug) this research is almost certainly less valuable than research on a breakthrough drug. It is probably reasonable to assume that somewhere in the neighborhood of half of the patent supported drug research is wasted on such duplicative research.

There is no obvious basis for projecting the extent to which software development would be advanced if all research was public and freely available to other developers. While the availability of more information would surely lead to better software that is developed more quickly, it is really only possible to guess at the extent of the gains from increased openness.

One further complication in projecting the gains in this respect from eliminating IPR protection is that it is not even clear how much software development currently depends on IPRs. While the Commerce Department provides estimates of spending on software (approximately $190 billion in the United States in 2005), much of this spending is for custom software programs that are tailored for the needs of a specific user (such as customer-service software for a commercial bank, rental-car agency, or online retail store). While the software developers may obtain IPRs for the software they install, most of the fees involved are for the design and installation of a firm-specific system. The lack of IPR protection would probably have little impact on the price of these systems or the ability of these software developers to charge for their work.

A crude way of projecting the amount of software development that depends on IPR protection is to use Microsoft's reported expenditures as a basis of calculation. In its quarterly statement for the second half of 2004, Microsoft reported that it spent $3.0 billion on research and development in the second half of 2004.[3] Assuming that this it spent roughly the same amount in the first half, this would mean that Microsoft spent a total of $6 billion on software development in 2004. If this sum is doubled to include the spending of other software developers, then it implies that the industry spent $12 billion of research supported by IPR protection in 2004.

Using this number as a starting point, it is possible to project the amount of wasted expenditures in software development due to IPR protection. Table 3 shows projections for low, middle, and high waste scenarios, making alternative assumptions about the waste due to unnecessary duplication and secrecy. In the case of unnecessary duplication, the low, middle, and high waste scenarios assume alternatively that 20 percent, 40 percent, and 60 percent of expenditures are devoted to duplicative activity that would not be carried through in the absence of IPR protection. The losses due to secrecy are assumed to be alternatively 10 percent, 20 percent, and 30 percent.

[2] This estimate is derived from the Food and Drug Administration's classification of new drugs and the pharmaceutical industry's estimate of the cost of developing duplicative, as opposed to breakthrough drugs, see "Financing Drug Research: What Are the Issues?"

[3] Microsoft's annual statement from the second half of 2004 can be found at [http://www.microsoft.com/msft/earnings/FY05/earn_rel_q2_05.mspx].

Table 3
Wasteful Software Development Expenditures Due to IPR Protection

	Waste Due to Unnecessary Duplication	Waste Due to Secrecy	Total Waste Due to IPR Protection
	billion 2005 dollars		
Low Waste	$2.4	$1.2	$3.4
Middle Waste	$4.8	$2.4	$6.2
High Waste	$7.2	$3.6	$8.6

Source: author's calculations, see text.

Clearly the projections in table 3 are little more than guesses. More research would be needed to generate an accurate estimate of the amount of software that is currently supported by IPR protection. It should also be possible to find some basis for providing better estimates of the amount of development that is wasted on unnecessary duplicative activity as a result of IPR protection. Assessing the relative rate at which software development might proceed in an environment of free and open software compared with IPR-protected software would almost certainly require guesswork in any case, because we will never see the two systems developing independently side by side. Obviously, there is a loss due to secrecy, but there is no obvious way of determining whether the numbers used in this table capture the range of plausible estimates.

With these extremely important qualifications, the assumptions used in constructing the table lead to scenarios in which between 28 percent and 72 percent of current software development expenditures are wasted due to the distortions created by IPR protection. The implication of such projections, if their basis turns out to be plausible, is that a system of publicly financed system of free and open software would require far less spending on software development to accomplish the same results. Alternatively, the same level of publicly provided spending could lead to far more impressive accomplishments, if free and open software were the standard.

Rent Seeking Activities of IPR Holders

Another major source of waste associated with IPR protection in software is the rent- seeking activities that it causes. The large gap between the price and marginal cost of the product gives producers far more incentive to advertise and market their products than would be the case if they were selling at their competitive-market price. While the additional sales that result from such marketing may be highly profitable for the producers, the expenses from sales efforts are largely a waste from the standpoint of the economy as a whole.[4]

No reliable data exist on the amount of money devoted to marketing relative to the amount spent on software development, but it is likely that the ratio is quite high. In the pharmaceutical industry, where patent protection leads to mark-ups that average 300-400 percent above the free market price, nearly twice as many people are employed in marketing than in research.[5] The relative size of the monopoly profits due to IPR protection in the software industry is even larger. Microsoft reported spending roughly 30 percent more on marketing than it did on research and development in the second half of 2004 (Microsoft 2005). If this pattern is repeated throughout the industry, then the calculations used in the last section would imply that approximately $15 billion a year is spent by the software industry on advertising and marketing.

While excess spending on advertising and marketing is probably the largest single rent-seeking cost associated with IPR protection, it is just one of many. A second potentially costly expense involves security measures that are necessary simply because the software is subject to IPR protection. At the most basic level, this involves designing digital locks, or other features of the software, that make it difficult to copy. Some portion of the resources devoted to software development must be employed to making the software more difficult to duplicate, instead of making it better for the consumer. Obviously, from the standpoint of the economy, such expenditures are a complete waste.

In addition to the resources devoted to designing software that is more difficult to duplicate, software producers also must make a wide variety of expenditures to ensure that their IPRs are respected. These expenditures includes efforts to monitor the distribution of their software over the web, legal fees associated with enforcing IPRs, and contributions to political campaigns and public relations efforts needed to sustain and extend IPR protection.

Beyond the expenses incurred by producers, there are also costs associated with enforcing IPR borne by other actors in the economy. These costs include paying for the law enforcement officials and the court personal required to enforce IPRs, enforcement efforts imposed on third parties such as Internet Service Providers and universities, and also restrictions on the developments of technologies that could facilitate the violation of IPRs.

This last category of costs is likely the most important. The software industry (along with the entertainment industry) has persuaded Congress to pass legislation that restricts the production of hardware or software that facilitates infringement on IPRs.[6] While the exact meaning of these restrictions has not yet been clarified by the courts, clearly they have the effect of slowing technological progress. If a hardware or software producer must worry that the courts will prohibit them from selling the product that they hope to develop, then there will be less incentive to pursue the development of technology

[4] Advertising and marketing can provide information about products, which is economically beneficial. However, if this information is misleading, and prompts consumers to opt for software that is less well suited for their purposes, the sales effort contributes negatively to the economy. Even if no resources were required for the sales effort, the economy would be better off without a misleading sales effort.

[5] (http://www.pharma.org/publications/publications/profile01/app_a3.phtml)

[6] This discussion refers to the Digital Millennium Copyright Act, which was signed into law by President Clinton in 1998.

designed to facilitate quick and accurate reproduction of digital material.

The set of costs implied by these efforts to secure economic rents are difficult to quantify. Software developers spend tens of millions of dollars on lobbying and campaign contributions every year.[7] They almost certainly spend an even larger amount on lawyers engaged in work intended to either protect their own IPRs or to defend the company against the charge of having infringed on the IPRs of others.

There are no good measures of the cost that IPR holders (more often in the entertainment sector than software) have imposed on third parties by requiring monitoring or other efforts to discourage IPR infringement. In some cases, that largest expense may be simply be the time involved. For example, if a college requires that 5,000 incoming students sit through a one-hour lecture on respecting IPRs, the implicit cost would be $50,000, if an hour of these students' time is worth an average of $10.

While it is virtually certain that search and copying software and hardware would be more advanced if the development of these technologies had not been impaired by efforts to enforce IPRs, there is no obvious way to measure the extent to which development of these technologies have been impeded. It is reasonable to assume that we would see substantially more progress in these areas if developers did not have to worry about legal actions from holders of IPRs.

In sum, there are clearly substantial costs associated with the rent-seeking of IPR holders in software. These costs are extremely difficult to measure, but nonetheless are likely to be substantial.

[7] The Center for Responsive Politics reports that Microsoft along spent more than $6 million on lobbying in 2004 (www.crp.org).

21st Century Alternatives to Patents and Copyrights

If there were no alternative mechanisms to finance software development, then there would be no choice but to live with all the costs that software-related IPRs impose on the economy. Fortunately, there are alternatives and there are good reasons to believe that these alternatives would be far more efficient than the existing system, saving consumers billions of dollars, leading to more jobs and economic growth, and the production of better software. The two alternative mechanisms discussed in this section are a system of direct government support, modeled after a proposal for an alternative to patent financing for prescription drug research, and a system of individual vouchers, which is an extension of an alternative mechanism to copyright for financing creative and artistic work.

The development of software is probably best supported by a mix of the two systems. Some types of software bear features that are probably more similar to pharmaceuticals, while other types of software have more in common with music, movies, and other creative work that are typically supported by copyright at present. Specifically, standardized software that is likely to be used by tens or hundreds of millions of people, such as computer operating systems, or word processing and spreadsheet applications, and provide a basis for many other spin-off applications, is similar to pharmaceuticals in that the assessment of quality can probably best be determined by experts in the area. While an individual consumer can assess the system that she finds most useful, she will not typically be able to assess the software for technical features, likes its ability to be adapted for additional applications or its resistance to viruses. For this reason, the allocation for funds to develop these types of software is probably best left to people who are experts in software development.

On the other hand, the quality of more narrowly tailored applications, such as video games, specialized graphic programs, or search programs is probably best assessed directly by the consumer. For these types of software, it would make sense to leave the allocation of development funds in the hands of consumers. Of course, there is no reason to draw a clear line between the two types of software. There would no harm done, if some of the funding intended for the latter type of software ended up promoting the development of spreadsheet applications, or some of the funding intended for the former type of software ended up being used to develop a graphic design package. The point of having two separate mechanisms is to ensure that no important areas of software development are neglected; no serious harm results if there are some areas of overlap between the two sources of funding.

IPR Alternative I: The Software Development Corps

As is the case with prescription drugs, the development of system software and widely used and standardized applications is probably best carried through by a central funding mechanism controlled by experts in the field.[10] There are several key principles that should be applied in setting up such a system. First is the open source/free software principle. The software developed through this system is being paid for with the public's tax dollars, this means that it belongs to the public. All software produced through this system should be fully available to the public to use and alter as they see fit (possibly subject to "copyleft" restrictions – anyone is free to use the software directly or as basis for future modifications, as long as they don't attempt to restrict its future use).

To enforce this principle of free software, it would probably be necessary to require that anyone receiving funding through this system be ineligible for IPR protection for any of their work for a substantial period of time (e.g.

[10] This model is based on a proposal for financing prescription drugs, which was part of a bill recently introduced in Congress (The Free Market Drug Act of 2004). A description of this mechanism can be found in "Financing Prescription Drug Research: What Are the Options?"

5 years). This would prevent developers from gaming the system by being paid to develop software, but then gaining IPR protection for their best ideas so that they could personally profit from them. The great feature of this system is that it would be largely self-enforcing. A copyright or patent granted to a person who received support through the Software Development Corps is simply invalid. Anyone can freely ignore their IPRs. The sanction to the person who has illegally claimed an IPR in such situations is simply that the courts will not enforce their patents.

The second principle that it would be important to preserve is competition. This can be accomplished by having the public funding for software development divided among several competing public corporations. For example, if the public funding for this mechanism of software development is $2 billion a year, then this money can be divided among ten firms that would each receive $200 million a year to support software development. These firms would then be free to use this money to finance the software research they deemed most promising, subject to the legal framework described above. The public corporations could either look to hire developers directly or subcontract out with firms or research institutions, or some mix of the two mechanisms. (The National Institutes of Health may provide a good model here. Less than one-fifth of its spending supports research that it directly undertakes, the rest is contracted out with private institutions and individuals.)

At regular intervals (e.g. 7 years), the work of the software development companies would be assessed by an independent commission of experts and users. The worst two would be shutdown, with two new ones being created to replace them. This should ensure that the companies feel pressure to use their funds effectively and that failed bureaucracies are not perpetuated indefinitely.

In addition, it would also be desirable to have a separate pool of money in this system (e.g. $100 million a year), which could be used as prize money to reward outstanding breakthroughs in software development. This should ensure that software developers have substantial monetary incentives. Such a prize fund could allow developers who do extraordinary work to get a bonus of hundreds of thousands of dollars, or even millions of dollars, if their work makes an exceptional contribution to the field.

Annual funding at the level of $2 billion a year ($2.1 billion including the prize money) would be sufficient to hire 20,000 programmers a year, assuming base salaries of $80,000 a year and 25 percent overhead.[11] This sum would likely be sufficient to replace the bulk of the software development currently supported by the IPR system. Such a commitment of funding would be relatively small for the federal government, being equal to approximately 0.08 percent of annual spending. Furthermore, it is likely to have most, if not all, of these expenditures reimbursed in the form of lower-priced computers and software. If governments at all levels (federal, state and local) buy 5 million computers a year, and the average savings on new computers and software came to $500 per computer, then the savings to the government would be $2.5 billion a year, considerably more than the sum committed by the federal government to support software development. (Of course, much of this savings would accrue to state and local governments.)

In this case, the benefits to the rest of the economy, in the form of much lower cost computers and software would be a free lunch – effectively the equivalent of a tax cut of between $80-$120 billion a year, with no offsetting cuts in government services. This would both lead to large immediate gains to consumers and to a substantial stimulus to the economy. The money that consumers save on buying

[11] The Bureau of Labor Statistics Occupational Employment Statistics Survey estimated the average annual wage of a computer programmer in 2004 at approximately $80,000 [http://bls.gov/news.release/ocwage.t01.htm].

computers and software can instead be diverted to other areas of consumption or may be used as savings to finance new investment.

The Artistic Freedom Voucher – Leaving Software Support to the Individual

The second mechanism that could be used to support software development is the Artistic Freedom Voucher (AFV), a system of individual vouchers.[12] Under this system, every adult would be provided with a certain amount of money (e.g. $100 dollars) which they could use to support any person they like who is engaged in creative or artistic work. Such work would include writing and performing music, writing or performing in movies, writing books, newspapers or designing software. The money could also be paid to intermediaries who support creative or artistic work (for example, an intermediary may specialize in supporting the writing of mysteries or the production of a certain type of video game).

In order to be eligible to receive funding through the AFV system, a person or intermediary would have to register with the government in the same way that a non-profit organization or a church must register with the government in order to obtain non-profit status. In such cases, the registration amounts to informing the government of what sort of creative/artistic work the individual does or the organization supports. The government does not assess the merit of the work, its only responsibility is to ensure that fraud is not taking place, in the same way that it verifies that religious organizations are in fact engaged in religious activity and not running a business or a tax scam.

It may be desirable to have a separate software development voucher, apart from a more general artistic freedom voucher, to ensure that a certain portion of this funding go to software development. The disadvantage of having this distinction is that it would make the system more complicated. However, it would be a possible modification if it turned out that individuals were neglecting software developers in their allocation of funds from an artistic freedom voucher.

As with the Software Development Corps, individuals who receive funding through the AFV system (either directly or through an intermediary) are prohibited from gaining IPR protection for their work for a substantial period of time after receiving this support. All the work must be placed in the public domain where it is freely available.

[12] This mechanism extends a proposal for supporting creative and artistic work described in "The Artistic Freedom Voucher: Internet Age Alternative to Copyrights"

The Competition Between IPR's and Alternative Mechanisms

It is important to note that the IPR system can in principle exist side by side in competition with these alternative mechanisms for supporting software development. There is no reason that software developers, like Microsoft, could not continue to produce IPR-protected software and try to sell it in the market. If they actually produce software that is sufficiently superior to justify paying the IPR-protected price, then consumers will still purchase it. These alternatives simply provide a basis for a market test of the two systems.

In this context, it is important to understand that this competition is between two forms of government subsidies, not between a government-funded system and a market-based system. In one system, the government pays for the development of software upfront either through the Software Development Corps or the AFV system. In the IPR system, the government gives software developers a state-imposed monopoly, where the government criminalizes competition for a period of time.

In a free-market system, software developers would simply be left on their own – producing their software and selling it as best they could. If another software developer had the ability to reproduce and sell the software, then the original developer would have no mechanism to prevent such sales. IPR protection is one way that the government interferes in the free market, in this case, in order to provide incentives to software developers. The appropriate question is whether IPRs are the best mechanism.

Conclusion

Relying on IPRs as the major means for supporting the development of software brings with a large cost in terms of economic inefficiency. The most obvious source of inefficiency is the enormous mark-up over the free-market prices that results from the government enforcement of monopolies in the sale of computer software. For the most part, software that currently commands substantial prices as a result of IPR protection would be available at zero cost over the Internet, in the absence of such protection.

However, IPR protection also distorts the process of software development itself. It encourages unnecessary duplication, as software developers devote time and resources to duplicating software that is protected by IPRs in order to gain a portion of the monopoly rents. In the absence of IPR protection, effort would generally be better directed towards improving existing software. Secrecy and restrictions on the use of IPR-protected software impede the process of software development as well.

IPR protection also leads to large expenditures by IPR holders to protect their rents. These expenditures include advertising and marketing, legal fees, and lobbying and campaign contributions. All ways in which IPR holders seek to maximize their monopoly rents. IPR holders also impose costs on others, such as Internet Service Providers and universities, by requiring them to take steps to enforce IPRs. In addition, IPRs have managed to obstruct the development of software and hardware that facilitates searches and reproduction of digital material.

The paper proposes two alternative mechanisms for funding software development, both of which can co-exist in competition with the IPR system. One would provide approximately $2 billion a year of government funding through a set of competing software development companies. In this case, all the software would be made freely available to be reproduced or altered, subject to copyleft principles. The second mechanism would allow all adults a certain sum of money (e.g. $100) which could be used to finance any creative or artistic venture of their choice, including the development of software or computer games. Any work supported through this system would also be immediately placed in the public domain.

It is quite likely that the government could fully recoup the money it paid for software development through lower prices on the computers and software it purchases. In this case, the lower computer and software prices seen by the general public would have the same effect as a tax break in the neighborhood of $80-$120 billion a year, with no offsetting cut in public services. Money that had been spent paying for computers and software would instead be devoted to either items, providing a substantial boost to the economy and leading to millions of new jobs.

Appendix

The calculations in Table 1 assume that the world sales for desktop computers were 80 million units in 2004. They assume that sales of laptops were 60 million. The average prices are assumed to be $800 and $1,000, respectively. It is assumed that without IPR protection, the average price of the computers purchased would fall by $200, and that the value of the additional software included with an average computer be $600 at current prices, with a value to consumers equal to $300. (In other words, it is assumed that the decline in the price of computers is equivalent to an average of $500 per computer.) It is assumed that the average price of a computer or video game is $29.50, based on data from the industry.[13] The low, medium, and high elasticity scenarios assume a constant elasticity of substitution utility function with elasticities of 0.2, 0.45, and 0.7, respectively. The estimates of current sales volumes are loosely derived from a recent article in the business press which put worldwide sales of notebook computers at 60 million units a year.[14] It is assumed that desktop sales are somewhat larger, at 100 million units a year. The estimate of the volume of computer and video games sales at 250 million units a year at an average price of $29.50 is taken from *Essential Facts About the Computer and Video Game Industry*, 2005, Entertainment Software Association.
[www.theesa.com/files/2005essentialfacts.pdf]. The calculations in Table 2 present sums from Table 1.

[13] http://www.theesa.com/files/2005essential facts.pdf
[14] These calculations assume 2005 world sales of notebook computers are 60 million, at an average price of $1000 per computer. This is derived loosely from recent public sales data ("
Taiwan took 81.3% share of global notebook PC sales in Q2" [http://english.www.gov.tw/index.jsp?id=12&recid=109038&viewdate=0]). The calculations assume that the 2005 sales volume of desktop computers is 80 million at an average price of $800.