Michael A. Cusumano

The Changing Labyrinth of Software Pricing

Exploring the malleable revenue pathways among the different varieties of pricing strategies.

The software business is an industry of frequent change. No changes are more complex, or important for customers and producers to understand, than the changes occurring in pricing, particularly for enterprise software.

Software vendors' revenues generally come in the form of three "buckets." Traditionally, the first bucket is the up-front license fee (which could reach millions of dollars for large numbers of users) for the "perpetual" right to use that version of the software. As a second bucket, software companies also sell to their corporate customers a separate maintenance agreement, which usually consists of an annual fee totaling from approximately 15 to 25 cents on the dollar based on the retail price of the licensed software product. As long as the customers continue to pay this annual fee and do not change the usage terms (such as to make the software available to more than the licensed number of users or the "named" users), they receive patches and updates to the product. The vendor might also include some technical support. Services required to install and integrate the software, train users, or customize the product constitute the third bucket of revenue for enterprise software vendors.

As I have discussed in earlier columns, the revenues of software product companies have been shifting from mainly up-front license fees (approximately 70% of the revenues of public software products companies listed on U.S. stock exchanges in 1990) to mainly maintenance and other services (just over 50% of these companies' revenues in the mid-2000s). It is not uncommon today for a software product company to have merely one-third of its revenues coming in from product license fees, another third from maintenance payments, and the final third from other services. Maintenance is treated as a service for accounting purposes because companies can declare these revenues only as the maintenance services—bug patches, product revisions, and technical support—are delivered.

JASON SCHNEIDER

At least part of the reason for this shift in software revenues from the product fee to services is that prices have been declining for enterprise software. The most extreme price decline, of course, is to *zero*—for "free" open source products like Linux, Apache, MySQL, Mozilla, OpenOffice (or its free equivalents), and a host of other commonly used programs. The price of software can and does fall to zero because the marginal cost of copying software or any digital product is essentially zero. Customers know this and have begun to rebel against paying high prices for copies of standardized software. Enterprise customers that buy a lot of software also know they can wait until the last week of the quarter or, better yet, the last week of the end of the fiscal year for their software vendor, and receive enormous discounts on the license fee—perhaps as much as 80%.

Another trend that is good for users, popularized by Microsoft, is the bundling of new products with an existing platform product or back-office stack of applications, with no incremental price increases. For example, the consumer saw a zero price when Microsoft bundled communications software and file management utilities with MS-DOS, or Internet Explorer and Media Player with Windows. It is the same when SAP or Oracle add supply-chain management tools or CRM applications to their basic set of applications and do not raise their prices. Companies bundle new features and whole new applications with older versions of their products to encourage customers to upgrade or continue paying maintenance fees.

Again, the trend toward "free" or zero pricing and declining license fees is beneficial for users but devastating for product companies—unless managers can figure out how to make money from services or convert users to different pricing models and delivery schemes (such as hosting) that make the real price of software more difficult to determine. In response to these trends, we are now seeing great creativity as well as complexity in the pricing of enterprise software and some consumer software.

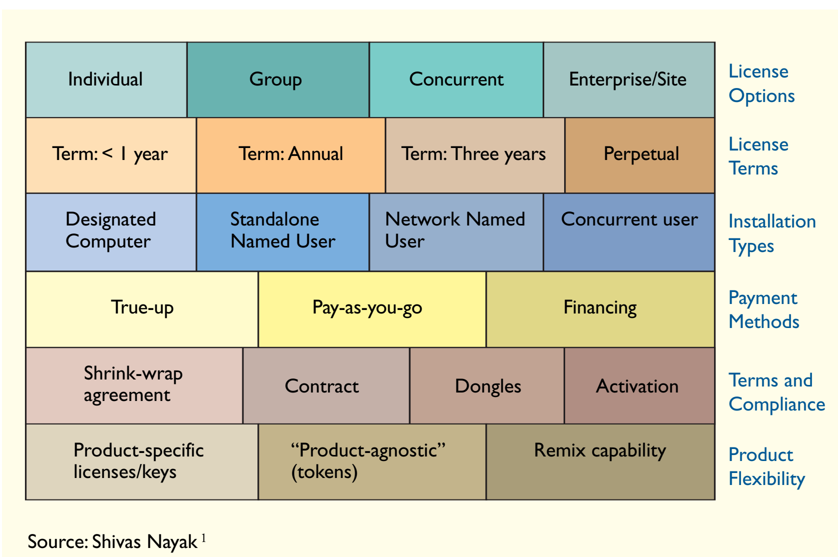
One adaptation is the adoption of a term or subscription license. In this pricing model, the customer buys the right to use the software product for a predetermined period, such as three years, and pays in installments. After the term is over, the customer either must renew the contract or cease to use the software product. Enterprise software applications, once installed in an organization, are difficult to dislodge, so most customers renew the licenses. As a result, this approach, while it seems to give more flexibility to the customer, primarily breaks up the large up-front license fee, combines this with maintenance, and allows the customer to pay in smaller payments spread over time.

A second variation of the term license is a very short term subscription, such as on a monthly basis. When combined with hosting, this concept is the same as "software as a service," popularized by Salesforce.com with its CRM product. Here, the software product vendor again bundles together the concept of a license fee with maintenance. Hosting gives complete control over the software to the vendor but also eliminates the need for the customer to pay for complex installation and integration expenses. Salesforce.com also provides professional services if requested, such as to customize the product or implement training programs.

A third variation is a usage license, as Oracle offers for its applications. Typically, this is a one-year license, with the fee based on some metric such as the amount of revenue handled by the software or the number of users.

In addition to these variations on the traditional software license, there are many variations used to determine the actual prices customers will pay (see the figure here). For example, the customer may license the software only for the use of particular individuals. But the license could also be for a certain number of concurrent users, group of users, users only at a particular location, or only for particular computers or networks of computers. The software license usually includes other terms for compliance with the specified agreement (such as hardware "dongles" or special activation codes that enable a user to use the software) as well as for payment.

Licenses can also let customers use any products in a bundle, or only certain products, or they allow customers to change the mix of products they are allowed to use ("remix").[1]



Attributes of software licenses. (Source: S. Nayak, "Pricing and licensing of software products and services: A study on industry trends," Unpublished M.S. thesis, System Design and Management Program, Massachusetts Institute of Technology, May 2006, p. 16.)

Despite the options, most customers continue to buy perpetual licenses. Usage or on-demand pricing appears to be relatively rare. AMR Research published 2005 survey data from 242 users showing that, for basic enterprise applications, 29% preferred the traditional one-time license fee, followed by 26% who preferred site licenses. Another 23% wanted on-demand or usage-based licenses and only 15% desired term licenses [1]. Macrovision recently published an even larger survey of nearly 500 software industry executives and more than 200 executives from users. They found that use of subscription pricing was actually declining, though vendors expected it to rise in the next several years [2].

In addition to this labyrinth of variants for the traditional enterprise software license, there are several variations of the "free, but not free" pricing model for consumer software and the occasional enterprise software product. One strategy has been called the "razor and blade" model—give away the basic software platform and then charge for other parts of the product systems as well as for updates or more advanced versions. Adobe is the best example of this model with its Acrobat product line—the company gives away Adobe readers but charges for the editing tools and server software. A second variant is the services model—give away the basic software product but make money on services such as installation, maintenance, training, and customization. Most companies that leverage free open source software products, such as Red Hat, follow this strategy. A third variant is the advertising model. Google is the best known company in this category, but Microsoft is prepared to move there as well with Windows Live and Office Live. Individuals do not pay for using functions like email, search, or even word processing and spreadsheet creation, although they must spend time watching advertisements paid for by organizations wishing to advertise.

What do all these changes mean? The traditional up-front license fee was simple for users and producers. It made a few companies rich and let them take advantage of the potentially 99% gross margins on reproducing software products. As prices of software products have fallen, the "99% of zero is zero" rule that I wrote about in *The Business of Software* has taken effect and forced many software product companies to close or sell to some larger competitor. Our database at MIT suggests we have "lost" nearly half of the public software product com-

[1]A useful discussion of these variations is a study I supervised, Shivas Nayak, "Pricing and licensing of software products and services: A study on industry trends," Unpublished M.S. thesis, System Design and Management Program, Massachusetts Institute of Technology, May 2006.

panies since their number peaked in 1997 at over 300. Product companies that have remained in the software business have had to adapt their strategies as well as pricing and delivery models. Product companies can no longer afford to spend enormous sums of money on R&D and marketing to build and sell features that customers do not want or use. Some data collected by another MIT student, Andreas Goeldi, suggests that monthly subscription fees with hosting may be the most common future strategy for enterprise software product companies with Web versions of their products. That would certainly simplify the current labyrinth of software pricing models for users and probably result in smoother, more predictable revenues for the software product companies that survive. C

References

1. Macrovision. Key trends in software licensing and pricing. Santa Clara, CA, 2006; www.siia.net/software/pubs/SW_Pricing_Licensing_Report.pdf.
2. Shepard, J. and Carter, K. *Software Licensing and Maintenance: What a Difference a Year Makes.* AMR Research Report. Boston, MA, 2005.

Michael Cusumano (cusumano@mit.edu) is a professor at the MIT Sloan School of Management and author of *The Business of Software*, Free Press, 2004.