

## Rules Of The Road: Valuing Commercialized Software-Based Technology

By David Drews and Dwight Olson

Companies that develop and own software often have the need to communicate the business value of their software in a meaningful way. The question “what’s your software worth” often arises in many different contexts, such as when acquirers/investors are interested in assessing their options, licensees desire a reasonable exclusive royalty/license price, or stockholders look to measure the company’s management performance, to name just a few. When considering the value context of software, it is important to include all appropriate elements of value in the assessment and include all the software product components.<sup>1</sup>

There are numerous methods for determining the value of commercialized software. These include general valuation methods, such as the well-known market or income approaches to intellectual property valuation, and software specific methods, such as the Constructive Cost Model (COCOMO and COCOMO II), the Software Lifecycle Management model (SLIM) and other cost-estimating techniques that focus on the efforts needed to replicate the functionality of the software in question.

All of these methods may be utilized throughout the developmental life of a software product, from conception and initial development to commercial launch and thereafter. This article discusses certain important elements associated with the application of these methods to software that may be misconstrued or incorrectly implemented. It is essentially a citation of the “rules of the road” when considering both the context and value of software, and points out elements of software value not usually considered in a typical IP valuation exercise.

This article discusses the integration of concepts from the Total Software Value-Ownership Value assessment [TSV(OV)]<sup>2</sup> analysis with other valuation techniques to provide a more comprehensive depiction of the values of software. While the TSV(OV) provides an estimate of the value of the software product from a replacement point of view, which is important in buy vs. build decisions, it also incorpo-

rates additional investments made to advance the software to the next stage of development/version, as well as identifying the additional software products infrastructure necessary to support the software for use. When combined with other valuation techniques, such as the Income Approach, which measures the potential value resulting from such efforts as licensing of the software, a comprehensive picture of the value of the software from many different perspectives will emerge. When valuing software, it is important to recognize all aspects of the asset in the analysis, including several that may be unique to software.

■ David Drews,  
IP Metrics, LLC, Principal,  
San Diego, CA, USA  
E-mail: [ddrews@ipmetrics.net](mailto:ddrews@ipmetrics.net)

■ Dwight Olson,  
V3Data, Principal,  
San Diego, CA, USA  
E-mail: [dolson@V3Data.com](mailto:dolson@V3Data.com)

The net present value of software is typically based on the potential income from future licensing cash flows. Historically, the user fees to license software are essentially independent of the cost and effort to create the software, and for PC software, this probably always will be the case. Otherwise, how can a few brilliant lines of code have a very high license fee in some applications, whereas a million lines of code that simply generate a report have a low license fee in other applications? A software product can be “squishy” in terms of the source of its value generation, and using traditional valuation methods, even those applied to other intellectual property such as trademarks, patents, and music copyrights, may give misleading results.

In general, the “Rules of the Road” for software valuation also include the following:

- Investors making decisions about the potential acquisition of a software company know that investments always have risk. So what are the unique risks associated with software? One risk is not compensating for the fact that software costs tend to grow over time, and therefore present novel issues not seen in other intangibles. For example, support costs to sustain software effectiveness occur throughout the lifespan of the software, and over time these costs

1. “Software & Valuation in the Information Society Part 3 TSV(OV),” *les Nouvelle*, June 2009.

2. *Ibid.*

may be many multiples of the original version development cost. In comparison, the maintenance costs for trademarks and other forms of patents and copyrights are seldom even close.

- Software licensees (or potential exclusive licensees) need to consider the costs represented by the opposing options of building competing software vs. licensing-in software. From economic theory, we understand that an investor will not pay more for a product than it would cost him or her to produce internally. As long as the revenue projections are accurate and the cost-to-build analysis encompasses all obsolescence and opportunity cost factors, and associated head-start risks are mitigated, the decision should be fairly straightforward. However, there are other, less obvious, costs and risks with the licensing-in option that need to be considered. For example, it may be necessary to implement custom modifications to a standard software product. These additional costs and time delays need to be part of the equation. As mentioned above, obtaining maintenance and support services for the software may be a significant cost exposure going forward, thereby reducing anticipated profit margins.

While some decisions may be based on cost, an understanding of the in-house capabilities for managing and mitigating these costs and risks is also important. We know that software maintenance and support is critical but how does one mitigate the risk of not overpaying the licensor for his contributions? What is the value of a future version vs. a current version? Who owns any modifications (and associated inventions) made to the software? A thorough understanding of all potential risks, costs and opportunities is vital to reaching the correct decision.

Also, any time that future cash flows and investment options are an important aspect of an analysis, one needs to consider the opportunity costs associated with any delay in bringing the product to market. In the most straightforward situations, these may include the interest lost on the funds invested in the development of the software. In addition, it may include current and near-term profits foregone on these products due to the decision to develop some aspect of the software in-house, which takes time, rather than licensing that capability from a third party, which could be implemented much sooner. If that additional development time pushes the launch of the product back six to twelve months, that lost time in the marketplace needs to be quantified and included in the overall value figure. This is another example of value in the form of avoided costs ac-

curring to the owner of previously commercialized software due to its relatively advanced stage of development.

- The value of an existing customer base should not be dismissed. First, depending on the type of software, the expenses necessary to create awareness of the software among the target market have already been invested for the most part. As that awareness grows and other elements begin to have an impact, such as word-of-mouth and positive reviews, the cost of each new additional customer drops dramatically. This reduced per-customer expenditure indicates that a larger percentage of future cash flows will fall to the bottom line. Also, as with the reduction in development costs, any expense that does not have to be paid points to an existing source of value for the owner in the form of higher retained cash flows.

- During an acquisition or investment involving software, how much of the property is disclosed via patents and copyrights, and how much is not disclosed in the form of trade secrets and other know-how in the transaction documents? The due diligence undertaken must include all valuable software components, know how, and intellectual assets. An important idiosyncrasy of software is that one must have access to both a physical manifestation of all of the components of a software product and the underlying rights to it in order to exploit it. Of course we have seen the emergence of technology escrow to begin to mitigate some risk but most often the escrow is not verified, validated, and valued. Thus many lenders and investors are attempting to mitigate short falls after the fact, oftentimes in the context of bankruptcy proceedings.<sup>3</sup>

- Companies that transfer the ownership of software assets to their foreign subsidiaries or contractors are oftentimes unaware of the value and extent of assets being transferred. Was it limited to the software patents listed in the re-assignment? What about the know-how, software tools, trade secrets, and unregistered copyrights that were likely key components of the transfer? When the valuation is related to a transaction, one must ensure that all assets necessary for the proper utilization of the target software product have been contemplated and included.

A major context of intellectual property and intellectual asset value is typically the present value of

---

3. "Leveraging Software via the Capital Markets," *les Nouvelles*, March 2008.

the income it will generate over its lifetime. This simple rule is the basis for most investment valuations (famous artworks being the classic exception), and estimating that future income is a major concern in investment decision making. It is certainly true that any method that requires predictions of future outcomes will never be precise, but having some consistency in defining the risk factors associated with the income stream may help in understanding the valuation analysis outcome. For example, before an investor should accept a software company's business plan financials, he or she should understand the following:

- 1) The size, cost and duration of future revenues and expenses associated with the development and commercialization of the software.
- 2) Size and nature of the market; competitive environment.
- 3) The various risk factors that will affect the likelihood of achieving the forecasted revenues and expenses.

For software, the size, cost and duration of future cash flows depends on many factors, including the revenues and expenses associated with licensing, maintenance and support. For licensing revenue, it could be as simple as the license price times the number of licenses. For maintenance and support, it could be a percentage of the current base paying for these annual services.

As for the impact of costs, consider this: when a new version of a software product is ready for release, selling licenses of the prior version will most likely be a money-losing activity. While the costs of copying and distributing software are typically very low, there is no benefit in continuing to license old versions. This is because supporting old versions of software will usually create a huge, ongoing cost burden for the owner.

Purchasing older versions of software does not make sense for the licensee either. It is likely to create huge expenses in adapting a prior version of software to the licensee's newer computer systems.

When determining the duration for future patent revenue, it can be as simple as determining the

remaining life of the patent, although obsolescence and competitive factors need to be considered. For software, it is never that easy. The remaining useful life of software is typically quite complex. The main concern will be the life cycle of the technology and its ability to transfer from, say, one Intel chip to another Intel chip, or from one version of an operating system to another. Also, the software technology's ability to scale from one user to millions will be an important factor.

For software, the size and nature of the market can be staggering or insignificant. In the U.S. alone, software licensed to users amounted to about \$120 billion in year 2000, about half of which was prepackaged software and the other half custom software. A good understanding of the market niche is required for applications that will compete against entrenched competitors. In addition, the number of potential computer-based systems upon which software will operate, be they mobile, server or desktop, is indeed staggering. For example there are over 1.4 billion PCs installed worldwide with a growth rate of just under 12 percent annually. PCs will likely surpass 2 billion units by early 2014.<sup>4</sup>

When discussing software, it is important to understand the risk factors that will discount the software's present value. Successful software products may have many versions and long lifetimes, but with corresponding high maintenance cost ratios over their lifetime. Some experts calculate the average software lifetime to be 10 to 15 years.<sup>5</sup> Many also believe that this average lifetime is likely to increase. For example, the UNIX operating system is over 40 years old and is still being used in many different versions, such as Red Hat. Some experts believe that the rate of software version update frequency is determined in part by the tolerance of users having to implement upgrades.

In the world of commercialization, all of these factors will have an impact on the valuation of software as well as software based patents. A comprehensive valuation analysis will, at a minimum, contemplate these issues, and will need to incorporate them if they are relevant to the subject situation. ■

---

4. *Personal Computing*, Mike Hanlon, July 1, 2008.

5. *Valuation of Intellectual Property and Intangible Assets*, 3rd edition; Gordon Smith and Russell Parr, Wiley, 2000, (p. 304).