Software & Valuation In The Information Society

Part Four: The Software Inventory Valuation—TOV (MV)

By Dwight Olson

The traditional valuation metrics that apply to physical technology governed by patents do not adequately work for the valuation of software not protected by patents but rather governed by privacy, trade secrecy, and copyright. Only a small portion of commercially available software is covered by patents. Copyright for commercialized software offers weak valuation principles. A recent article in les Nouvelles, discussed that the accounting rules required by FASB do not accurately apply to software valuation.[1] This article will review other traditional valuation methods used to value technology and show that these valuation methods are also inadequate for the business valuation of software. Since the value generating principles such as identification, possession, ownership, and trade secrecy for software can be lost easily by failure to take the appropriate due-diligence measures, some best software valuation practices will be highlighted such as the need for an IP audit to give assurance that the value principles of ownership, possession, trade secrecy and privacy are being maintained.

A software product is a complex bundle of assets and consists of both intellectual property and intellectual asset components. As discussed in a *les Nouvelles* article regarding software inventory valuation "one could consider the components of intellectual property (IP) to comprise patents, trademarks, copyrights and industrial designs because these four intellectual properties have legislation to govern, protect, and permit value propositions for the legal owners."[2] The other components or intellectual assets (IA) would then be considered to include trade secrets and know-how or the codified, tangible descriptions of specific knowledge which a software business uses to support commercialization.[3] Thus software, by virtue of its component inventory[4] (or software inventory), can be the subject of various valuation strategies. For example, if there are patents (IP) involved then the statutory rights granted can be used to value a given patent, to generate monopolistic income via a license program, to allow strategic value activities such as cross licensing, or to perform an investment IP management analysis.[5] When patents do not apply, software still has value propositions and those are value propositions central to the intellectual assets.

This part four article looks at software valuation constructs when there are no patents involved or when the values of the Intellectual Property Right (IPR) are not of consequence independently of the software product. This is usually the case for most software products and when software is developed for use exclusively inside an organization. Many of these constructs can also apply to Web sites, open source and to open collaboration software based projects, however, it may be questionable if the constructs apply to software provided under the *free* open source software (FOSS) community. The constructs best apply to software that has been developed for commercial purposes or internal use and could theoretically be licensable or leveraged to return value to the owner.

Value Concepts

According to some experts in the field of intellectual property valuations, there are four main value concepts used while valuing any IP[6] and they are appropriate for software: *ownership* value, *market* value, *fair* value, and *tax* value. Ownership value is the view of value if the owner is deprived of the ability to monetize and/or exclusively use.[7] The concept of market value assumes that if a comparable property has fetched a specific price then the property in question would fetch a price similar to it. Estimating fair value of an asset assumes that the transaction is not going to take place in an open market where the free forces of supply and demand are active, so in estimating fair value the owner and purchaser are

[1] Software Valuation Part 2, *les Nouvelles*, December 2008.

[2] Intellectual Asset Identification, The First Step in an Intellectual Property Management Program, Dave Tyrrell and Gary Floyd, Vertex Intellectual Property Strategies Inc.

[3] Software Inventory Valuation Part 3—A TSV (OV), *les Nouvelles*, June 2009.

[4] See Tables 1 and 2.

[5] See *http://intellectualassetsinc.com/client-services/ma-analysis/.*

[6] Valuation of Intellectual Property Rights, Ms. Namita Chandra, Paras Kuhad and Associates.

[7] *Ibid.*

conceptually brought together where the deal is fair to both parties. The tax value concept relates to valuation for financial reporting and tax purposes.

For tax and financial book purposes, FASB 86 was undertaken in response to requests by the Securities and Exchange Commission (SEC) and the American Institute of Certified Public Accountants (AICPA) to be able to record certain costs of software as a capital asset. As discussed in Software Valuation Part 2, many of the basics of the final rule are not the appropriate constructs for ownership, fair or market values of software for business purposes. However, the risk factor constructs outlined in the exposure draft developed prior to the ruling, but were not included in the final rule, are still very appropriate for software valuations for business purposes.[8] This early framework included the risk factors of technology, market, financial, management commitment, [and ownership statements for intellectual property rights] feasibility. FASB 86 did not specifically call out ownership statements for intellectual property rights as a risk factor, but certainly, if you don't have the work for hire agreements or the copyright assignments for the ownership of the intellectual asset components, then you really take on a risk to monetize or value software you don't own.[9]

Business Purpose Value Methods For Market And Fair Value

The various methods employed by professionals who analyze IP for market and fair values are *cost*, *market*, and *income* based methods. These three methods are the basics in valuing intellectual property assets and the experts recommend multiple valuations to reach a final determination of market or fair value.[10] An introduction to these methods follow with a discussion of how some specific software constructs should be included. According to the experts, although each of these methods of valuation has limitations, together they can provide a set of very useful decision making tools.[11]

Cost Based Valuation And Software Considerations

A cost based valuation model focuses on the costs incurred to develop the intellectual asset. It provides an estimate for the value of the asset that is tied to the cost to create or acquire the ownership of the asset. For software asset valuations it is important to remember to identify and cost all software product components, see Table 1,[12] using both financial input as well as cost models. Reasonableness equations should be used to bias any financial information relative to the actual cost for developing the technology.[13] Many variables associated with developing software, such as the number of lines of new and modified code, were found to be useful in developing reasonableness equations.[14] A basic concern when considering any cost is its reasonableness. A statistical examination should be conducted, to develop a central value and an acceptable range of values for a software product cost estimate. The cost estimating of software projects evolved from the works of Walverton[15] attempting to correlate the size of effort in person-months with the size of the product expressed in Thousand Lines of Source Code (KLOC). Large dispersion of data forced Walston and Felix to take into account additional variables such as programmer experience and the complexity of the application as well as over twenty variables.[16]

■ Dwight Olson, CLP,
LESI Copyright Committee,
V3Data, Principal,
San Diego, CA, USA
E-mail: dcolson@ ix.netcom.com

An issue in software valuation analysis is the mistake of thinking that an end-user licensed proprietary product or even an available open source software product is always a factor in replacement valuations. Such consideration may be tendered in analysis consideration for valuing software for internal use, but should not be considered in the case for value considerations in investment transactions. If the analysis is to determine the fair value or market value where one side of the transaction believes there is the opportunity of making money with the software

[8] FASB 86 "Exposure Draft proposed that the costs incurred internally in creating a computer software product would be charged to expense until cost recoverability had been established by determining market, technological, and financial feasibility for the product and management had or could obtain the resources to produce and market the product and was committed to doing so."

[9] Software and Valuation in the Information Society, *les Nouvelles*, June 2008.

[10] Valuation of Intellectual Property Rights, Ms. Namita Chandra, Paras Kuhad and Associates.

[11] *Ibid.*

[12] *Ibid.*

[13] *Ibid.*

[14] R. W. Wolverton, "The Cost of Developing Large-Scale Software," IEEE Trans. On Computers, June 1974.

[15] *Ibid.*

[16] C.E. Walston and C. P. Felix, "A Method of Programming Measurement and Estimation" (IBM System Journal Vol. 16, No. 1, 1977).

by whatever business model, then “owning” and possessing the intellectual assets, software inventory, is very different than being a licensee (a software end user license) of the commercialized software represented by the software intellectual asset components. Caution should be taken if valuing software that falls under the FOSS GNU General Public License especially version 3 (GPLv3)[17] and similar free communities. A best practice in all cases would be to include an IP/IA software validation audit looking for free open source components, unknown copyrights, third party software to get an IP legal opinion as to the risk and ability to monetize or carry out revenue generating activities.

In the standard cost method for IP, such as a computer hardware patent, one normally takes into account the physical depreciation and functional obsolescence of the hardware technology represented by the patent in calculating the replacement cost. At issue for software valuation is that software does not have physical depreciation as does the tangible hardware upon which it resides. Software should be looked at in terms of technology feasibility (which should include functional obsolesce) not depreciation. Methods used to depreciate tangibles as well as intangibles are based on the assumption that the goods being valued lose value over time. Such depreciation schedules are based on wear, the loss of value due to obsolescence, or changes in customer preferences.[18] However, well-maintained software, in active use, does not wear out, and is likely to gain value.[19]

Current thinking regarding software life cycle analysis is from the current accounting standard based upon the waterfall model first described by Winston Royce in 1970, ironically as an example of a flawed model for software development. The waterfall model is a sequential process, in which the life cycle is seen as flowing steadily downwards like a waterfall through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, [Commercialization], Maintenance[20] and then Obsolescence. Today, as probably for the past twenty years, new software is almost always put into use with only partial functional requirements fulfilled, and then evolves over time, frequently, over very long times, as the software is enhanced, customized, adapted as new technical requirements and standards are discovered and changed.[21] One of the greatest inherent values of software is its ability to evolve and adapt to new challenges through modifications, revisions and continuous improvement.

It is important in biasing the ownership/market/fair value by the appropriate software remaining useful life (RUL) factor that is appropriate for its “level of abstraction layer.”[22] One factor of RUL is the software’s ability to transcend the hardware or system technology upon which it currently operates. Software has been developed over the past twenty years with the design to migrate to or include “new” hardware or system technology. It is here where a subject matter expert can provide the appropriate software RUL “level of abstraction layer” factor. A specific example would be for a software driver that is developed to manage a particular hardware device has a limited, or small, level of abstraction factor and its remaining useful life is usually the same length as the tangible hardware.[23]

Experts agree that the cost of an intellectual property asset to the owner, whether the asset was created or acquired, is a cost that cannot be used to price an intellectual property asset. This is also true for software as the cost to develop may have little to do with the price for an end-user license. However, the ownership value vs. an available replacement end-user license should be adjusted by freedom to operate or proprietary advantage factors in investments related to internal-use-only software.

Another issue for software ownership is that of reuse. The issue was discussed briefly in Software Valuation Part 2, as a reuse factor of modules where a program module may be used more than once. For example, in a software program a module may be reused in the program to process different attributes. This may be a significant factor when a software component in the inventory may be used as a component in another product for a completely different market. This may also be an important factor in looking at

[17] The Free Software Foundation, use the GNU General Public License for most of its software; it applies also to any other work released this way by its authors.

[18] What Is Your Software Worth?, Gio Wiederhold, Communications Of The ACM, September 2006.

[19] Spolsky, J. Joel on Software. Apress, 2004.

[20] Wikipedia reference for “waterfall model,” *http://en.wikipedia.org/wiki/Waterfall_model.*

[21] Software Valuation Part 2, *les Nouvelles*, December 2008.

[22] *http://en.wikipedia.org/wiki/Abstraction_layer.*

[23] The *ISO-OSI networking model* comprises seven abstraction layers and the Open Systems Interconnection Reference Model (OSI Reference Model or OSI Model) is an abstract description for layered communications and computer *network protocol* design.

fair value analysis especially if the ownership of the software is not in question, and at stake is establishing a fair value for an exclusive license to use the commercialized software in a particular market.

Experts contend that if an intellectual property asset is no longer useful to the owner, the owner should sell the asset at any price regardless of whether the price covers the owner's costs. In the case of IP (patents and trademarks) this may well be the case, but little is known about software intellectual assets in these distressed or non-useful owner situations. Maybe a better solution for non-used or owner failed marketing attempts of the software could be for the owner to place the software in the open source world. This would allow for the situation that there really may be a need and, the software may have a new life through collaboration. It is important to remember that there are many business monetization models in the open source world and many times the right open model for monetization does not come into focus until after acceptance by an appropriate community of users. However, the strategic motivation for putting software into a *free* open source software community should be well understood before doing so.

Market And Income Based IA And Software Considerations

The article Software Valuation Part 3[24] discussed constructs and risk factors for ownership valuation of the software inventory or TOV (OV), these constructs and risk factors should be kept in mind for TOV (MV) or market or fair value analysis. Market-based valuation models for patents and trademarks estimate the value by looking to the marketplace. Patents or trademarks that are comparable to those in question are identified, and the sales or transaction revenue actually derived from those comparable assets are used as an estimate of the value of the patent or trademark asset. When comparable IP assets can be readily identified, market based valuation models are relatively easy to apply, and can yield useful projections. Finding comparable transactions is the key to the market valuation method. This is relatively easy in the case of real property where the neighborhood, square footage, and number of rooms can be used to compare past home sales in order to value a potential home sale. Such is usually not the case for IP such as patents and trademarks as there are few public trading markets for patents or trademarks (say other than Ocean Tomo[25] and a handful of other international IP auctions), the terms and conditions of IP asset transactions vary widely, IP assets are inherently dissimilar, and the details of IP asset transactions are rarely available to the public. If it is difficult to find a comparable for a patent or trademark, it will be even harder to find a comparable for a software intellectual asset.

A central issue for software inventory and trade secret valuation is the notion of registry. Unless software's intellectual assets are identified and registered (private[26] or federal[27]) by the owner, valuation is quite hypothetical. Software intellectual asset components are normally considered to be unpublished copyrights or trade secrets and without a private or federal registry[28] of a clean deposit for identification, any market or fair valuation is certainly questionable. Without the trade secret, copyright, or knowhow registered somewhere, verified, and audited, what did the valuation refer to? In the case of a patent asset, all information and data regarding the patent is deposited and registered in the PTO, as is a trademark. A best practice for both software and trade secret valuations is to have the asset components placed in a software escrow account then verified, validated, audited and then valued, see best practices below. Depositing the software component assets in the escrow account by the valuation subject matter expert is also a good way of providing an audit trail of what was valued. A best practice for an open source software valuation should be to register the components in a private registry as any member of the open source consortium could change at any time any of the components in the open source database, such as "Sourceforge."[29]

Just as for IP, factors that affect the software market comparability include: the relative balance of power between the buyer and the seller, industry concentration, market size, barriers to market entry, the growth outlook for products incorporating the commercialized software components to be valued, and anticipated new product introductions. Experts

[24] Software Inventory Valuation Part 3—A TSV (OV), *les Nouvelles*, June 2009.

[25] Ocean Tomo introduced the world's first public auctions of patents, trademarks and copyrights.

[26] *www.dsiescrow.com* Data Securities International started the first software escrow company in 1982.

[27] *http://app.legalis.net/geneve/gb/indexgb1.htm* A Swiss registry associated with the EU.

[28] The Agency for the Protection of Programs (APP) is a European association of software developers and information technology professionals, registered under the French law of 1901.

[29] *http://sourceforge.net/ SourceForge*.net: Open Source Software.

in the field of valuation of IP asset have identified four basic requirements for the market method to be functional. They are:

- An active market must exist for the asset;
- There must be a sufficient number of similar asset exchanges in the recent past;
- Price information on similar asset exchanges must be available to the public; and
- The exchanges must be between independent parties.

To complicate the matter for software valuations, few software ownership transactions are ever available to the marketplace and even when software is included in a company acquisition, seldom are the software owned assets apportioned from the acquisition price.[30] What we typically find are paid licenses to commercialized software products, which have little to do with the ownership, fair, or market value of company owned software.

Some Additional Income Based IA And Software Considerations

Income-based valuation models for IP make use of current and forecasted future revenues to develop an estimate of asset value and are true for software. Under this valuation model, the amount of license revenue that a commercialized software product can generate is used to establish the software product's value. The IP models adopt a forward-looking perspective, estimating future earnings that can be derived from commercial use of the IP. Different companies and valuators apply different definitions and projections regarding "revenue forecasting." As a consequence of this diversity, the income-based valuation model differs, in practice, from company to company.[31] This is especially true for software income valuations.

Income-based models function best when there is accurate information to support future income and future cash flow projections. This information is more likely to be available when the IP asset is similar to one already in the marketplace or if the asset will reach a clearly established market. Thus, income-based models are less effective when income and future income information is sketchy or speculative. Experts suggest that there are four parameters that need to be looked into for estimation on the income method. They are:

1. The amount of net income the asset is expected to generate.

2. The time period over which the income is expected to be received.

3. Determination of the present value discount rate for future income.

4. The risk of realizing the future income.[32]

In estimating the net income inherent in software, one must estimate how much income the software will generate during its life, which in most cases also requires estimating its (RUL) life factors. If the software is licensed to others, the expected income depends on the licensing revenue or number of licenses times the license price. Then to value the net income also requires the risk of realizing future income. Future income is directly tied to the ability of the software to meet future obstacles. A majority of software costs are incurred during the period after the software is first delivered to the marketplace. These costs are primarily due to software enhancements, support and maintenance. This refers to the activities to preserve the software's market value by modifications, revisions and continuous improvement in functionality and performance in the marketplace. Successful software products have many versions, long lifetimes, and corresponding high maintenance cost ratios over their lifetime.[33] Software lifetimes are typically 10 to 15 years and even longer and are likely to increase.[34]

The last parameter or discount rate for future income is commonly built on future cash flow estimates associated with a particular asset. These models project future earnings and expenditures attached to the asset. Those estimates are also discounted to account for the time value of money and the uncertainty as to the accuracy of the projected cash flow. The net present value of the future earnings is calculated so that the estimated potential value of the asset can be compared with similar estimates.

At issue for software in analyzing future income is determining software operational risk. Software operational risk is the risk factor associated with migration to future technology, and issues determining systemic problems such as internal procedures and failures by people and systems to continue to deliver expected support.

[30] See FASB 141 and 142.

[31] Valuation Of Intellectual Property Rights, Ms. Namita Chandra, Paras Kuhad and Associates.

[32] *Ibid.*

[33] What Is Your Software Worth?, Gio Wiederhold, Communications of the ACM, September 2006.

[34] Smith, G. and Parr, R. Valuation of Intellectual Property and Intangible Assets, 3rd Edition. Wiley, 2000.

As mentioned, many different analytical models are presently being applied to estimate the market or income value of intellectual property. I would call the reader's attention to the model proposed by the article "What Is Your Software Worth?"[35] In the article, Gio proposes an analytical model using many of the constructs discussed in this article. Although this method has its limitations, it provides an alternative to investors and financial managers to better understand the potential market or income value of software.

A typical weakness common to all software valuations is the failure to account adequately for the legal aspects of software's intangible assets, their development, protection, and transfer. To account for those legal aspects effectively, the valuation models should include estimates or actual costs associated with a legal audit and opinion of the ownership, possession, and other rights in any market or fair value analysis. This audit should go hand and glove with any encumbrance audit of licenses for the software.

Central Constructs For Software Valuation

The central constructs for software valuation are the identification of the software inventory components, possession by the corporation, and statements of ownership to the components. These are the primary constructs to keep in mind for software valuation. Another important point to remember in valuation of a software product, is that only after the market has been established and demand is apparent, the software inventory contains the only assets with value.[36] Then after the market is established current and future revenue of the software product will again be dependent upon these same intellectual asset components. In other words, quality of the software inventory will play an important part in fair and market valuation of software.

Where Is Software Valuation Used?

1. Acquisition Financing–In many business acquisitions, the value of computer software is an important consideration in guaranteeing either cash flow-based or asset-based financing. In some tax cases, the valuation of computer software may directly impact the refinancing or recapitalization of an established business, outside the realm of an acquisition or transaction.

2. Debt Financing–In many situations the tangible assets such as receivables, plus intangible patents and trademarks are not sufficient in value to cover or secure many investment transactions. This may be for LBO's or smart investors who want additional security. Some investors may wish to lock assets into the company vs having them walk away should bankruptcy conditions occur.

3. Business Sale/Purchase–The valuation of computer software plays an important role in the overall valuation of an enterprise. Obviously, an overall business valuation is an important consideration in the determination and negotiation of a business purchase or sale price. Plus, because computer software is typically a depreciable asset under FASB 86 for federal income tax purposes, its value may have a greater impact on the overall business value than other assets which may not generate tax amortization benefits, such as going concern value of goodwill. See also FASB 141/142.

4. Management Stewardship–The valuation of computer software and its impact on the overall business valuation may be an important consideration on the periodic assessment of the effectiveness of management's stewardship of software business assets.

5. Intercompany Transfer Pricing–In many firms, computer software is developed by a centralized headquarters staff, but these internally developed systems are used by the various divisions and subsidiaries. The appropriate intercompany transfer price for the use of software should be a direct function of that asset's value. Depending on the relative locations of the firm's branches and subsidiary operations, the transfer price for the use of internally developed computer software can have both Federal and state income tax implications.

6. Gifting Programs–In taking advantage of the annual gift tax exclusion, family business owners can transfer beneficial interest in the firm's computer software to various family members. Based upon a type of licensing arrangement, the company will retain the use of the software while the owner maintains direct equity control.

7. Estate Planning–In many cases where computer software is a material asset of the business, periodic software appraisals will help determine the value of the business equity and the value of the business owner's estate. Even if the business owner does not implement gifting or other wealth transfer programs, these periodic appraisals will allow the business owner to plan for long term tax considerations.

Software Best Practices For Ownership/Fair/ Market Valuations

1. Identify the software product components to

[35] Gio Wiederhold, Communications of the ACM, September 2006.

[36] Software Inventory Valuation Part 3—A TSV (OV), *les Nouvelles*, June 2009.

be valued using Table 1 and Table 2. Then assemble a valuation team. Expertise in software valuation, software escrow, licensing and IP/IA law (depending if there are significant patents and/or trademarks to be included, more than one legal person may be required), should be assembled for the various audits. If the valuation is for tax or SEC purposes, then a tax accountant should be included on the team.

2. Identify the context of the valuation. The premise of any valuation engagement can affect its conclusion. So depending on the particular purpose of the analysis is to determine a context for:

a. a "sale" or transfer,

b. a LBO,

c. a M&A value,

d. an exclusive license transaction value,

e. an investment value process,

f. damage claims,

g. insurance purposes,

h. and then how is the valuation going to be used?

i. by the venture capitalist,

ii. by the acquiring CFO,

iii. by the investor,

iv. by the licensee,

v. by the IRS or SEC?

The actual amount of value may reflect different considerations for ownership cost as well as projected costs, cost if acquired for license, cost if acquired for exclusive use, or the value of first-to-market advantages. In any case, it is important to know that software has economic value even if use is only internal. A software fair value amount that seems reasonable in a hypothetical arm's-length negotiation may not be indicative of what later might be assessed for a market value. Should this be for the SEC or IRS, a tax accountant should be included in the team assembly.

3. Make sure that there is a deposit into a private registry (escrow) for the trade secret, know-how, and unpublished copyright items identified in Table 1. Unless software is identified and registered in a private registry (for some open source products it may be practical to use a federal registry for the source code) by the owner, valuation is quite hypothetical. Software intellectual asset components are normally considered to be unpublished copyrights or trade secrets and without a private or federal registry of a clean deposit for identification, any market or fair valuation is certainly questionable. Trade secret deposit into the private registry (a federal registry is not recommended by this writer) should be transferred using the "public" encryption key of the private registry for security purposes.

4. Perform a validation of the escrow registry components looking for non-proprietary components and code including an IP legal audit. The IP audit should look at the employment agreements, work for hire, purchased 3rd party software, and any associated copyrights and license agreements. If there are associated patents, an analysis of the possible conflict with open source should be examined. When managed correctly, by legal counsel and executive management who set policies regarding use of open source and 3rd party proprietary code, then valuation of the software product is possible by providing a more accurate valuation. Assuming the open source or third-party components can be detected or extracted from the software inventory, then using the appropriate software inventory valuation factors for ownership and validation can be used in fair and market valuations. In those situations where patents are involved to provide freedom to operate or dominate positioning, caution should be observed in any patent valuation that may also be involved. In the district court case, States Court of Appeals for the Federal Circuit, 2008-1001, Robert Jacobsen, Plaintiff-Appellant, Matthew Katzer,

Table 1. Software – Intangible Asset Inventory

Marketing and Sales	License Management
Marketing plans and collateral	DRM and license controls
Client Support Systems	Back office system
Installation and training	R&D Systems
User documentation and help	Internal Design Documentation
Client databases	Source code with comments
QA and testing	Source code control with comments
Bug/support system	ASP databases
Testing code and data	QA test and procedures
Manufacturing System	3rd Party software
Specific build guides	Open Source & Strategy
Commercialization Strategy	
Product plan (release and updates)	

Table 2. Software IP Bundles

Marketing and Sales Copyrights	Trade Secrets and Know How
Marketing collateral	Design documentation
Patent(s)	Source code with comments
Defensive and offensive	Formulas
Product Copyrights	Process know how
Executable code	Operating platforms
User documentation	Manufacturing instructions
Installation instructions	Configuration data
GUI's	QA test and procedures
Domain Name(s)	3rd Party Software
Licenses	SAAS or ASP databases
Encumbrances	Client databases

Defendants-Appellees, Jacobsen's operative complaint against Katzer/Kamind included not only his claim for copyright infringement, but also claims seeking a declaratory judgment that a patent issued to Katzer is not infringed by Jacobsen and is invalid. This of course has devastating implications for any patent values associated with the software product. For companies that have already taken steps to comply with the open source licenses to which they are subject, the decision in Jacobsen should not have a significant impact. Jacobsen has the potential to significantly increase the risk of noncompliance with open source licenses. For those companies that have elected not to comply with open source licenses or, as is the case with many companies, have chosen to remain unaware of the open source software licenses to which they may be subject, Jacobsen should be all the incentive that is necessary to adopt and implement a sound open source license compliance program.

5. Perform a deposit verification audit of the escrow registry components. This should encompass all of the components identified in Table 1. The verification should be at least to the level 2. Level 2 verification was initially developed by Data Securities International and the escrow industry to help identify source code for each and every component (file) that is part of their software package(s). An internet reference to a European level 2 verification document[37] is included for reference.

6. Perform a software ownership valuation. A Total Software Value (TSV) is the overarching principle guiding valuations for software assets and intangibles. The goal of the ownership valuation process is to give software owners an overview of the likely range of the values of a software product as a function of the view of value if the owner is deprived of the ability to commercialize, sell or exclusively use. The ownership value provides a foundation for further valuations, such as fair or market valuations, and exclusive licensing or acquisition discussions for monetization purposes. Items 1 through 5 should be completed prior to this analysis.

a. The following will be helpful if not in the deposit registry:

i. Copies of any relevant business plan, marketing study, financial statements and independent appraisal.

ii. Descriptions of any litigation, past or present.

iii. Copies of any contract, licensing agreement masters, or offers for exclusivity.

iv. Available economic data on the industry in which the software is used.

v. Copies of promotions and advertising materials used during the past year relating to the software or the product in which it is incorporated.

vi. Cost information relating to the existing or proposed software product including cost accounting records and/or feasibility studies.

b. If there are patents involved, the valuation team should carefully read the patent and software product description and then review or interview a patent attorney, a subject matter expert in the technology, and an IP licensing professional. The result should be a set of evaluation opinions that are appropriate for the patent and the software product. A Legal opinion, Subject Matter Expert evaluation of the patent for purpose rating, and an IP management analysis using biblio-metric tools.[38] These opinions will be biased any economic value determined and based upon the purpose of the valuation (see Item 2). The purpose should be for carrot licensing situations or transactions, not for calculating damages in stick licensing. If a quick patent valuation is needed, then using a patent valuation tool such as the Patent Value Predictor may be helpful in establish early guidance. "Pat-

[37] *http://www.escrow.nl/downloads/Verification%20Procedure%20Level%20II.pdf.*

[38] Such as Intellectual Assets Inc. IP portfolio analysis.

ent Value Predictor: Patents are a right to exclude others from making, using, or selling a product or service covered by the claims of the patent. The Patent Value Predictor models this situation by assuming a substantial fraction of GDP is covered by all patents, and then estimating the fraction of the GDP covered by each patent using sophisticated data analysis and additional modeling based upon macro economic data and financial data. Patent Value Predictor models the profit associated with a patent to be the fraction of the GDP covered by the patent (i.e., the nominal sales of product that our model predicts to be covered by the patent) times the profit margin. From that information Patent Value Predictor obtains the annual profit protected by the patent. Patent Value Predictor calculates the current value of the patent to be value of the annual profit for the estimated remaining term of the patent."

c. FASB 86 risk factors and software RUL factors will also bias the ownership value

7. Fair value best practices. Determine the buyer's motivation for fair value consideration and determine the amount of bias of the ownership value. Consider the applicability of market or income valuation approaches over ownership value. Ideally, all three approaches should be considered. However, many valuators prefer the income approach for valuing unique, income-generating properties especially if patents are involved. A cost approach is seldom useful if only patents are to be valued, and the market approach may not be relevant because patents are unique by definition and comparable patents may be difficult to identify.

a. Do an income-approach valuation. It is important to separate patent valuation from software valuation if patents are involved. For patents, variations of an income approach to valuing patents can be broadly classified as royalty-based or profit-contribution methods. "Profit contribution" refers to the profit attributable to a patent. "Royalty" refers to the income stream expected by the patent holder under a licensing transaction. Whether a profit contribution or royalty method is more appropriate in a given situation depends upon the premise of value or expected future operations—for example, whether a patent is to be sold separately, used for reserve, or commercialized as part of a going concern. For software, the income approach will be used to look at the revenue derived by licensing the software times the number of licenses less the cost to maintain market share. See also Software Valuation Part 3 for patent value with software as the commercialization component.

8. Market value best practices. Determine the motivation for market value consideration and determine the amount of bias of the ownership value. Again, consider the applicability of market or income valuation approaches over ownership value. Ideally, all three approaches should be considered. A market approach is seldom useful if only software is to be valued, and the market approach may not be relevant because software is unique by definition and comparable software products may be difficult to identify.

9. The software valuation report. The report(s) should document the software valued, assumptions made and limitations of the analysis. It should also contain the procedures followed and the conclusions reached including the qualifications of the valuation team members and their respective contributions. If an essential team member is missing—for example, if a subject matter expert did not participate in the evaluation process—the report should disclose the possible impact of that absence as a limitation of the valuation analysis. ■