State of practice in software valuation

Software is everywhere, but putting a value on it is a tough challenge. A distinguished panel of experts discuss how it might be done

By **Bill Elkington**

More and more major businesses and industries are being run on software and delivered as online services – from movies to agriculture to national defence.
Marc Andreessen, venture capitalist and co-founder of Netscape

Perhaps the single most dramatic example of this phenomenon of software eating a traditional business is the suicide of Borders and corresponding rise of Amazon.
Andreessen

Software is like entropy. It is difficult to grasp, weighs nothing, and obeys the second law of thermodynamics – ie, always increases.
Norman Ralph Augustine, former CEO of Lockheed Martin

Constant revolutionising of production, uninterrupted disturbance of all social relations, everlasting uncertainty and agitation, distinguish the bourgeois epoch from all earlier times. All fixed, fast-frozen relationships, with their train of venerable ideas and opinions, are swept away, all new-formed ones become obsolete before they can ossify. All that is solid melts into air, all that is holy is profaned, and men at last are forced to face with sober senses the real conditions of their lives and their relations with their fellow men. Karl Marx

Software is one of the fundamental means by which we innovate. We use it to eliminate jobs and make enterprises more efficient and more competitive. We use it to put greater power into the hands of individuals, so that they can have more power over the means of production, rather than the means of production having more power over them.

For example, here is something that Roger McGuinn – founder, lead singer and lead guitarist of The Byrds – has said about software: "Now, you can just get a laptop, get some software, put a microphone on it and make a record. You have to know how to do it. It does help if you've had 35 or 40 years of experience in the studio. But it still levels the playing field so artists can record their own stuff."

From what I have been reading about 3D printing, there may be a parallel with music. After all, with a laptop, some software and a 3D printer, you should be able to manufacture quite complex plastic and metal goods on your own in the not-too-distant future, which gives the individual worker power over the means of production he never had or hoped to have in Marx's day, or even Mao's. Some sectors of manufacturing could shortly return to a cottage industry setting, where it was before the Industrial Revolution, but with vastly increased productivity. And all because of some very sophisticated 3D modelling and design software, a high-speed processor and 3D printing technology.

And of course, the reason that all of these resources can be put in the hands of the individual – the creative worker – is that they are cheap. The elements we are talking about have been made by market forces to be ridiculously cheap, the software included. No longer is the worker at the mercy of a few rich people holding most of the capital. No longer is he at the mercy of huge corporations for a job. The

IAM *is the official strategic media partner of the Licensing Executives Society (USA and Canada) Inc (LES). As part of this wide-ranging cooperation agreement, both parties work together to provide thought leadership relating to the strategic creation of value from intellectual property. With this in mind, every issue of* IAM *will feature an exclusive article written by an LES member or members that focuses on an IP value creation issue. The content and authorship of the article is always agreed in advance with the* IAM *editorial team and is subject to the same editing process as all other articles published in the magazine. The work and the opinions of the author(s) are personal to the author(s) and do not represent those of LES or their respective employers.*

The panel

The distinguished individuals debating software valuation with Bill Elkington are:

- Dwight Olson, a principal of V3Data, Deposix USA, director of licensing for EXEN Holdings and business adviser for CONNECT. He is a past president of the Certified Licensing Professionals Inc, a former president of the Licensing Executives Society (LES) (USA and Canada) Inc and chair of the LES International (LESI) Endowment and IP Valuation Committees. During his 40-year career in the business of computer software commercialisation, Olson founded Data Securities International, the world's largest technology and software escrow company, and was involved in the start-up of several software companies. He has also served on numerous company boards and continues to write and speak extensively on topics of technology escrow as a business tool, valuation of software and trusted third parties in licensing.
- Sri Divakaruni, director of innovation business development at United Technologies Corporation (UTC). His team's mission is to leverage UTC's US$4 billion of annual R&D investments and 17,000 patents into strategic alternate markets and applications. Avenues to IP monetisation include technology and patent licensing, as well as joint development. Prior to joining United Technologies, Divakaruni was a business development executive at IBM. During his 24 years with the company, his assignments spanned IBM's corporate technology and IP licensing, research, hardware and services divisions. He holds an MS in electrical engineering from Rensselaer, NY and a BS in materials science and metallurgical engineering from the Indian Institute of Technology, Chennai, India.
- Joy Murray is worldwide director of the Microsoft Business Solutions (MBS) business desk at the Microsoft Corporation. Her team helps the field sellers around the world of MBS to close and structure software licensing transactions around the company's products. Prior to her position in MBS, Murray was the director of IP licensing in the IP licensing group at Microsoft, where she developed and managed significant licensing programmes for the company, including technologies that were line of business, mobile phone and email related. Prior to joining Microsoft nearly nine years ago, Murray worked in Boeing's commercial aircraft sales division, as well as its IP licensing organisation. Her early career was spent working for Boeing and for local software start-ups in technical, sales and marketing, management and various licensing positions.

creative worker – the cottage musician, entrepreneur, software developer, investor, travel agent, industrialist – has the tools to create a living for himself; and through the internet – through software – he has access to world markets.

Software permeates everything

This software-created revolution in the economic power of the individual is matched by what software is being made to do for the corporation. Companies' competitiveness has been transformed through the use of software frameworks; software modelling, development and test tools; business models built on providing software rather than hardware products, providing software as a service or providing information through software applications; software systems that manage many of the enterprise's operations processes; software collaboration tools that effectively integrate the work of global design and development teams; and so forth.

Marc Andreessen is right: software is eating our traditional businesses, in addition to creating whole new businesses. It is the locus of contemporary capitalism's "Creative Destruction", a term coined by Joseph Schumpter, which was inspired by Schumpter's reading of Karl Marx. It is how the corporation is innovating how it does business and how it innovates what it is in business to do.

In product company after product company, the value of the product is in the software that is at its heart. In company after company, application-specific integrated circuits are being morphed into software that runs on general purpose processors. In company after company, technology trends and market forces are forcing the product software to be abstracted from the company's product hardware. The hardware, for the most part, in embedded products is being commoditised. The differentiation is in the software. Thus, the computer has become a tool for our imagination, with software the language of our imagination.

The automobile industry is a good example of some of these trends. One reads that 40% of the value of a new car today is in its electronic systems. One reads that a typical car has 40 to 50 controllers and microprocessors, and that luxury models contain double this number. One reads that a typical new car today runs about 10 million lines of code, with projections as high as 30 times that number in about 10 years, to accommodate the greater and greater automation of the car's various systems, reaching substantially into the complete operation of the automobile itself – the "driver-less car" that Google and others are promoting.

If Andreessen and Norm Augustine are right, a great deal of the real substance and value in capitalist economies and capitalist companies and capitalist products and services is to be found in the software. Software permeates everything.

Valuation challenges

But how should we value software? Much is written about patent valuation, but what about software, the intangible stuff into which a great deal of tangible functionality is being converted at a frantic pace? What about software, that essential medium of complex corporate processes? What about software, that inexorable force for both job destruction and value creation. That force for both efficiency and creativity? For both corporate destruction and venture creation?

How do those of us in postmodern

corporations who manage intellectual property and who provide expert guidance on IP value actually do software valuation? How should we do it? How do we and should we think about its value? How do we guide others around us to think about its value? When we share our software with our business partners, what value should we be extracting? What are the metrics we should use? What is the relation between patent value and software value to a corporation swimming in a sea whose medium is software? Whose food is software? Whose offspring is software?

Is the essence of software value as airy as that of patents? Or is there something to software that one can get one's hands on? Is it somehow more substantive? More understandable? More comprehensible? Is its value more demonstrative? More objectively derivable?

For this issue's roundtable article, I have lined up three very interesting experts on the value of software. Their backgrounds are diverse and their experience is extensive. They have much to say that is instructive and insightful. Let's listen to what they have to say with the understanding that I and they are expressing our own personal views that do not necessarily reflect the views of our employers, our employers' customers or clients. They are: Dwight Olson principal of V3Data, Deposix USA, director of licensing for EXEN Holdings and business adviser for CONNECT; Sri Divakaruni, director of innovation business development at United Technologies Corporation; and Joy Murray, worldwide director of the Microsoft Business Solutions business desk at the Microsoft Corporation.

In industry after industry, the trend is to put more intelligence and functionality into software. Inevitably, industry partners – customers, suppliers and co-development companies – will want or need this software, even when the software development company does not identify itself as 'in' the software business. For such companies, how do you recommend they think about valuing such software?

Dwight Olson (DO): The intangible assets owned by any software development enterprise include the technical knowledge (and know-how) of its staff; the competence of its sales force; the business knowledge (and experience) of its management; its goodwill and reputation (including trademarks); the value of its intellectual property (including trademarks, copyrights and patents); the commercial value of its licences; and the value of its software inventory. When buying complex technology, companies must look at the total cost of the project, which consists of hardware, software and services.

But valuation of the software, as such, is not an easy task; and the simple use of a net present value (NPV) calculation to achieve a value using the income method to the software component/product based on revenue streams may be most difficult. There are many reasons for this, such as the inability to associate revenue streams back to the software components or internal processes. As a result, my recommendation to many companies is to do a cost analysis using the replacement method.

A cost-based valuation model focuses on the costs incurred to develop the software 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 using both financial input as well as cost models. Reasonableness equations should be used to bias any financial information relative to the actual cost of developing the technology. 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.

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 attempting to correlate the size of effort in person-months with the size of the product expressed in thousand lines of source code. Large dispersion of data forced Walston and Felix to take into account additional variables such as programmer experience and the complexity of the application.

However in cost modelling, Kemerer suggests that a distinct disadvantage of any formal model is the inconsistency of estimates. He conducted a study indicating that estimates varied from as much as 85% to 610% between predicated and actual values to complete the project. For costing analysis, he suggested that calibration of the chosen model can improve these figures; however, formal models still produce errors of 50% to 100%. He goes on to say that: "One of the most important objectives of the software engineering community has been the development of useful models that

Sri Divakaruni, director of innovation business development at United Technologies Corporation

“As the smartphones wars illustrate, event-driven valuations (ie, threat of infringement or lawsuits) provide opportunities for monetisation, but little insight into valuation methods”

constructively explain the development lifecycle and accurately predict the cost of developing a software product. To that end, many software estimation models have evolved in the last two decades based on the pioneering efforts of the above mentioned researchers.” These developments have added greatly to the potential use of such models to predict more accurately fair market values – especially in those cases when the software product has not achieved market penetration.

Sri Divakaruni (SD): Such companies should consider their ‘non-core’ software as a monetiseable asset that can be licensed. The licensing income stream can then be re-invested into ongoing R&D or selling, general and administrative expenses, to offset expenses. Depending on the breadth of software developed and value to licensees, this can also be a smart way to expand the company’s portfolio and/or presence in alternate markets. That is, licensing of software can be an arrowhead to introduce the company’s core offerings to other markets.

A common concern expressed is that the company does not intend to get into the software business and cannot provide Level 1 through 3 support as a typical software vendor/licensor. While an obstacle, this should not be a show stopper. Rather than licensing to end users, the business model to pursue in this case would be to license to software service providers who in turn would install, customise and maintain software to their end clients.

Joy Murray (JM): First and foremost, we all know that the true value of anything is what someone will pay for it. However, let us set this aside for a minute. For companies that do not identify themselves to be ‘in’ the software business, I would encourage them to look at valuation first by how they intend use or license it to the company that ‘is’ in the software business. There are many elements to take into consideration: will they maintain ownership of that intellectual property? How will they license it (exclusive, field of use limitation, non-exclusive)? How many other companies are they likely to want to license it to, taking into account the opportunity in the market and its lifecycle? Compared to other markets, the software market has a relatively short lifecycle and so time to market and lifecycle are critical factors. The target market and demand are also big factors in valuation.

In your experience, how do companies that identify themselves as ‘in’ the software business differ in their approaches to software valuation from companies that do not identify themselves that way?

DO: I think that this question would be better answered by companies that are ‘in’ the software business. However, in my past experience in ‘in’ companies, little was done by anyone to assign value to a software asset. Applying management responsibility based upon value principles was simply not done. Maybe Sarbanes-Oxley (SOX) and Financial Accounting Standard (FAS) 141/142 will begin to change that.

The idea that management is judged based on whether the software asset value is increased or decreased under particular management decisions is non-existent. Software valuation has its basis in the valuation of the asset component or inventory.

Software-based intellectual assets are known in the computer industry as the software inventory. This software inventory could be considered to include both intellectual property and intellectual assets. One could consider traditional intellectual property to comprise just patents, trademarks, copyrights and industrial design registrations, because these have legislation to protect the legal owners in the disclosure of these assets. Non-disclosed intellectual assets in the software inventory then could be considered to include trade secrets, unpublished copyrights, know-how, source code, databases or the codified, tangible descriptions of specific knowledge which a business uses to support commercialisation.

Trade secrets, unpublished copyrights and know-how are those software inventory assets that contain valuable proprietary information that belongs exclusively to the business, and that are used in the business to provide economic or competitive advantage. Many proprietary software vendors consider trade secrets to include unpatented inventions, formulae, processes, devices, patterns, designs, design drawings, source code, customer databases and internal operations manuals.

SD: Fundamentally, the former typically do the due diligence of defining how their software fits as part of their overall corporate strategy; while to the latter group, software is a means to an end whose tangible value to the company is very hard to quantify. To elaborate on types of due diligence done, they develop answers to questions, such as: is this the software core to the value I bring to my clients or non-core? Does it provide me with a competitive advantage?

What does my SWOT (strengths, weakness, opportunities, threats) matrix look like? Can I maximise re-use across other platforms? How best do I protect my intellectual property in this software?

JM: My experience has shown me that companies in the business have a much longer view in valuation just by nature of owning the roadmap and also being responsible for a return for their shareholders.

Do you have a view as to the relative usefulness or accuracy or validity of candidate software valuation approaches? If so, would you mind sharing that view?

DO: I believe that the method of valuation is best suited to the appropriate context. The various methods employed by professionals who analyse intellectual property 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. According to the experts, each of these methods of valuation has limitations, but together they can provide a set of very useful decision-making tools. For example, if one is looking at an M&A opportunity or investment opportunity into a company, then the income method will be used by the acquiring or investing organisation and it is best if the software company understands that. A savvy software company should know the market value using an NPV calculation of the income stream and the factors used in this method. However, the software company may want to attempt to get a valuation price higher based on increasing the value by using positive asset factors for adjustments – that is, discussing the current patent portfolio and how it could protect the market position. Then again, using a discussion of the trade secret valuation may provide unique value positioning and income protections.

The income method may be appropriate in licensing situations where there may be exclusivity in the licence. However, use of income in early-stage software development is problematic. I know that most financial folks will disagree and want to do an income projection with questionable forecasting factor. In my opinion, when you are looking at early-stage software, the income streams are not appropriate no matter how well you think that someone can forecast. Likewise, for software use licensing (assume non-exclusive), the cost replacement method and value might lend credibility to a specific royalty or licence fee discussion with the licensee.

SD: Most software valuation methods centre on trying to assess future income streams. In my opinion, there are two key factors that need to be discussed to provide an informed response.

The first is context. When Google makes 90% to 95% of its revenue from search, what is the relevance of judging valuation approaches to its Android or YouTube platform? Similarly, the reason that there is such a wide range of opinions and estimates on the 'value' of Facebook, Amazon or other social mobile companies is the uncertainty that comes with judging paths to monetisation, stickiness of platforms and loyalty of end consumers. So for the scope of this exercise, we are not going to discuss software where there is a fundamental disconnect between what the software does and the value that it provides to the end consumer.

The second is the value proposition to end clients. Of all of the methods for valuing software intellectual property, the most useful and most readily embraced are those that value it in the context of economic value created for the consuming client (deriving risk adjusted economic value is even better, where probabilities can be assigned to litigation and risks associated with workarounds can be quantified). As an example from healthcare, a drug formulation that addresses gastrointestinal distress in pigs has a quantifiable economic value to the hog farmer who defines a market acceptable price. However, if this same formulation is determined to prolong the lives of human cancer patients, it justifiably commands a much higher market price, even if the cost of making the formulation has not significantly changed.

This frames my bias that methods of valuation based on estimating development costs or on counting lines of code are essentially worthless. As the smartphone wars illustrate, event-driven valuations (ie, threat of infringement or lawsuits) provide opportunities for monetisation, but little insight into valuation methods.

JM: The approaches that I have seen, like Monte Carlo methods and so on, are all good pulses on the value. However, when the rubber hits the road, the value is what the market will bear. It is very much an exercise of negotiation. The software valuation approaches often provide you with data to determine your starting place of negotiation.

Dwight Olson principal of V3Data, Deposix USA, director of licensing for EXEN Holdings and business adviser for CONNECT

"Currently, I know few specialists who have commercialised software products and are now involved in IP valuations and value outside of the patents and trademarks that apply to the software"

In your experience, do best practice companies do software valuation once the software is developed or once the software requirements are in place? Whichever is the case, what is the rationale?

DO: My concern is always understanding the context of and for the valuation. If this is an early-stage situation and no revenue or income is real, then forecasting is fine for business management decisions because management will control the opportunity to capture a market and future income stream. But to attach any value other than a replacement value plus trade secret value at early stage is really stretching it, in my opinion. Once revenue streams can be determined and cost factors applied, value principles for the asset are appropriate using income methods. The hard part is the factors that adjust the value. I am not talking about the price of money, but the reality factors that are in the forecast. Forecasting factors (not the cost of money) are more in line with evaluation principles than with valuation method factors. The concept of linking evaluation to valuation methods is just starting and few have applied any thought to this outside of patent portfolio evaluation/valuation principles defined by the value of the asset to the organisation.

SD: Best practice companies do prospective software valuation, but they almost certainly do some combination of the two. This is not necessarily bad. With markets and requirements changing, having a feedback loop with real data from clients is invaluable. (I do not envy the folks responsible for pricing Research In Motion's BlackBerry 10 or Microsoft's surface platforms in their rapidly changing world!).

Another factor for consideration is whether the primary revenue streams for the company are from software (eg, Microsoft, discounting its Xbox gaming platform or SAP), or whether software is an enabler to other revenue streams (eg, IBM, where software is less than one-quarter of its revenue, but generates nearly half of its profits). A sale of a Websphere or a Tivoli licence to a large account from IBM often accompanies sales of accompanying servers and storage, besides signings for IBM's business and IT consulting services, giving IBM more flexibility in pricing its software in the larger context of value provided to its clients.

JM: My experience is that typically a company would value the software development effort prior to developing it. Any investment must be rationalised and software is no different. An anticipated ROI on a software roadmap helps management to prioritise development. This ROI typically includes market demand and market share.

In companies that are 'in' the software business, are there typically software specialists who work with the product teams, or are software valuations performed by business generalists or individuals in some other discipline? What are the benefits or issues that result?

DO: I think that this question would be better answered by companies that are 'in' the software business. However, in my experience in surveying the LES societies on valuation methods, few organisations perform 'ownership' valuations. The programme managers who put together business plans containing financial data are the closest that I am familiar with. I am not sure of the value of this exercise beyond budgeting. But as I already mentioned, perhaps SOX and FAS 141/142 will begin to change that.

SD: The pricing teams in software businesses typically tend to have a solid understanding of their domain, competitors and SWOT matrix, and they work closely with the lines of business on software valuation.

JM: What I have experienced is that there are software engineers and developers, and often R&D areas that develop the roadmap and products. R&D teams often have business development experts who perform inbound IP licensing. In companies that are in the software business, there are pricing specialists who assess the market, competitors and general market strategy and who create the pricing stock-keeping units for the business.

In companies that do not view themselves to be 'in' the software business, are there typically software valuation specialists who work with the project teams, or are software valuations performed by business generalists or individuals in some other discipline? What are the benefits or issues that result?

DO: Again, I think that this question would be better answered by companies that are not 'in' the software business. However, in my experience in surveying the LES societies on valuation methods, few organisations perform 'ownership' valuations. Probably the programme managers who put together business and product plans containing financial data are the closest that I am familiar with. I am not sure that, outside of the budgeting process, there is value in this exercise for any potential 'outside' stakeholder.

But once again, maybe SOX and FAS 141/142 will begin to change that.

For example, if we look at market-based valuation models for software patents and trademarks to estimate the value, then patents or trademarks that are comparable to those in question are identified, and the sales or transaction price actually derived from those comparable assets are used as an estimate of the value of the software 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. However, finding comparable transactions is the key to the market valuation method. This is relatively easy in the case of real property, where the neighbourhood, square footage and number of rooms can be used to compare past home sales in order to value a potential home sale. But it is usually not the case for intellectual property, as there are few public trading markets. Instead, 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 component.

SD: The norm in this case appears to be software valuations performed by business generalists or individuals with expertise in other disciplines. While there may be objectivity in making unemotional, unbiased evaluations, there is real danger in being unable to appreciate how 'tangible' the value of the software asset is in the context of the overall solution. The problem gets even more complicated when the software being valued is not marketed as an end solution, but is embedded as part of an integrated hardware plus software solution; or even worse, when the software is primarily developed for developing tools and methods solely for internal use.

JM: What I have observed is that these companies typically rely on third-party specialists which can range from third-party valuation experts to outside counsel at law firms. Over time, they often develop this skill set internally.

What are the challenges to getting knowledgeable software valuation experts to play a meaningful role? How do you recommend that these challenges be overcome?

DO: This is a very loaded question. I'm not sure how you get knowledgeable software valuation experts to play a more meaningful role. Currently, I know few specialists who have commercialised software products and are now involved in IP valuations and value outside of the patents and trademarks that apply to the software. This is usually because the client only asks for the patent value. And few have brought in the idea of evaluation of the patent as a premise to value. In my opinion, we need to begin also to get a handle on valuation of trade secrets, databases and other components of the software inventory to really make headway. We focus too much on protection and not on balance in content. For example, Karl Jorka suggests: "Trade secrets are the 'crown jewels' of corporations; they are the IP of the new millennium." Once we get this, it will certainly be a start.

SD: Software valuation is often relegated as a problem that is too difficult to solve by those who are not in the software business – more art than science. While it is true that software has certain attributes that are very different from most other capital-intensive industries, this creates just as much opportunities as challenges. While software seems under siege from open source companies or companies providing software as a service, how many other industries have the luxury of having such high gross margins? Besides a maniacal focus on technology leadership, this underscores the need for leaders to truly understand the markets they serve and drive to making software valuation a priority, since it can have a significant impact on the company's bottom line.

JM: I think that one of the main challenges for the software valuation experts is just getting their foot in the door and developing relationships at the decision-maker levels at software enterprise companies; and they build the trust and respect within the enterprise, helping them to understand the market, competitors and the business within tight time constraints. Not all companies have the luxury of time in educating a valuation expert – especially when time to market is critical.

When software valuation is done well and makes a meaningful contribution to the enterprise, what does that look like? Please tell us a best practices story.

SD: While this may rile some readers, Apple did a good job preparing for a defence of its seven smartphone patents from a software valuation standpoint. However

Joy Murray, worldwide director of the Microsoft Business Solutions business desk
"For companies that do not identify themselves to be 'in' the software business, I would encourage them to look at valuation first by how they intend use or license it to a company that 'is' in the software business"

controversial as to whether those patents should have been granted in the first place, it did a good job of defining the economic loss to its business and winning an eye-popping US$1 billion-plus award that was clearly far greater than what it invested in developing the code behind the seven patents.

JM: It takes into account the general market, market share and current position in the market; competitors; customers' perceived value; and timeline of value. You definitely know it was accurate when it is well received by the market. Perceived value matches market expectations.

When software valuation is not done well and/or does not make a useful contribution to the enterprise, what does that look like?

JM: This is actually pretty straightforward, in my opinion: it doesn't make a useful contribution when you don't learn anything from it and the expectations do not meet the expected outcome.

For the enterprise to achieve best practices in software valuation and software pricing, what needs to be done?

DO: Oh my goodness! Here are some thoughts for M&A best practices for a software company with software products in the marketplace.

First, identify the software components to be valued, including all intellectual property and intellectual assets; then assemble a valuation team. Expertise in software valuation (and validation), software registry escrow (with verification expertise), licensing and law (depending on whether there are significant patents and open source to be included, more than one legal person may be required) should be assembled for the various audits and steps chosen. If the valuation is also for tax or Securities and Exchange Commission purposes, then a tax accountant should be included on the team.

A central issue for software inventory and trade secret valuation is the notion of registry. Unless software intellectual assets are identified and registered in escrow by the owner, valuation is quite hypothetical. Software intellectual asset components are normally considered unpublished copyrights or trade secrets, and without an escrow registry of a clean deposit for identification, any market or fair valuation is certainly questionable. Without the trade secret, copyright or know-how registered somewhere, verified and audited, to what did the valuation refer? A best practice for both software and trade secret valuations is to have the background asset components placed in a software escrow account verified, validated and then valued.

For mergers, acquisitions or investments in technology companies, two additional processes should be considered prior to the valuation process: verification of the deposit and validation of the registry deposit. The verification process analyses the deposit to determine the nature of the deposit contents. For example, the standard software escrow industry today provides for multiple levels of verification to give reasonable assurance that the registry contents contain the information that is the technology to be valued. Validation of the deposit works in conjunction with the IP audit (legal process) to provide details of the deposit ownership.

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 is that after a market has been established and demand is apparent, the software inventory contains the only assets with value. Then after the market is established, current and future revenue of the software will again depend upon these same intellectual asset components. In other words, quality of the software inventory will play an important part in establishing a fair or market price for the acquisition.

SD: A few best practice recommendations are outlined below.

Define software value in the context of economic value derived by your clients. Economic value to a client can increase top-line revenue, operating cost reduction, time to market improvement, improved productivity or efficiency and reduced defects or maintenance costs, to illustrate a few examples or even combinations of some of these factors.

Comprehend all revenue streams for smart pricing. Since software is easy to replicate at negligible cost and additional units sell for a lot more than incremental cost, having an accurate and defendable picture of anticipated revenue streams is critical to optimised pricing. Licensing software developed for a specific industry into alternate verticals can often be a lucrative source of IP monetisation, without being a threat to the 'core' business that the company is in.

In the field of software valuation, is it useful to make a distinction between ‘value’ and ‘price’? Please explain.

Dwight Olson: Oh my goodness, yes! In my opinion it is very important to keep value and price separate. The concept of value has many ‘values’, depending on the appropriate contexts. I consider price to apply when a software asset changes hands or is licensed – that is, the price of the asset to be the agreed upon price for the sale or the agreed upon price for the royalty/licence/subscription fee for use. The value of the asset is in the context. The ownership value is the replacement value, plus other factors, of the software asset. The market value context might be the value that the software asset might achieve from the income stream associated with the software. Then there are the fair value considerations that might apply in tax considerations. Thus, to me, price is associated with a transaction.

Sri Divakaruni: Price is one of many dimensions that reflect economic value to a company. To illustrate with a few examples: Google may appear to have paid an extraordinary premium price for Motorola, but as one contemplates the spin-offs of Motorola’s set-top box business and the IP arsenal that Google accumulated to defend itself from IP litigation, besides giving itself a real chance to take on Apple by leveraging Android into smartphones and tablets built by Motorola, one can see economic value (partially reflected in its enterprise value) that well justifies the price paid. For an even more illustrative example: Microsoft developed a focused strategy through the past decade to leverage its patent portfolio not to extract licensing fees, but to build a reputation as a trustworthy supplier of open source software and platforms such as .NET. The economic value derived from settling dozens of lawsuits far outweighed the price of walking away from additional licensing revenue streams. You can read more on this in Marshall Phelps’ book, *Burning the Ships*.

Joy Murray: In my opinion, value and price are two different species completely. Value is what someone believes something is worth and is a result of perception of benefits. Price is the amount set for the value. For example: “The value I received from taking x class was understanding y, meeting z etc; but I paid abc for the class.” Both are connected, but are two different concepts – at least to me.

Action plan

Senior IP management people in a companies and industries that are being ‘eaten’ by software should consider the following:

- Whether their company has adequate processes in place for software valuation prior to pricing.
- Whether the available market data is sufficient to provide a reliable input to the valuation work.
- Whether the release process for software is consistent with the need for valuation and pricing reviews with the appropriate business leaders.
- Whether the appropriate business leaders have an adequate understanding of software valuation methods and their applicability to pricing.
- Whether software valuation approaches have been sufficiently well systematised to provide predictability and efficiency in the valuation and pricing process.

Consider synergies across marketed, embedded and internally used software to accurately comprehend costs and revenue potential.

Consider maintenance costs over the lifetime of the product. While legacy software can be a cash cow (think mainframes and software written in COBOL/ADA still in use in the financial or healthcare industry), maintenance costs can often exceed upfront development and testing costs over the life of programs. Thankfully, a combination of upfront installation and annual maintenance is the norm in most industries, which helps to define a win-win relationship with clients.

Consider lower upfront pricing of software in conjunction with royalty streams, especially where your software is part of a larger system sale by your client. Besides aligning interests and building goodwill with your clients, this can provide for annuity-like revenue streams.

JM: First of all, enterprises in the software business are likely to have the valuation skills internally, unless they are a young start-up or are entering a new market. Hiring a valuation expert with market specialties can be valuable depending upon the company’s internal expertise and available resources.

There are many issues here and many dimensions to the questions I have asked. Software valuation looks one way when one is doing an M&A transaction. It looks another way when one is valuing a product. It is different once again if one is talking about ancillary tools and frameworks and models.

Embedded software will need to be looked at differently from software that is not embedded. Software in a business-to-business transaction that involves licensing of intellectual property into non-strategic markets will look different from a software development kit that is provided by a platform company to individual developers. Applications that run on smartphones will look different from applications that run on servers.

Safety-of-life software systems will look different from game software. Source code will be handled differently from executable code. Patented software will need to be treated differently from unpatented software.

Jurisdictions matter, particularly when patents are involved.

Meeting the challenge

How should we advise the CEO of General Motors or John Deere or State Farm Insurance or Marriott when it comes to software valuation best practices? How should they be thinking about the value of their existing software and their future software in their software licensing activities with their business partners and with their customers?

How do we reduce the complexity of all this down to some easy-to-understand recommendations as to how to think and behave and price their product and service offerings and their ancillary software tools and frameworks and models rationally rather than whimsically? Horizontal markets – markets in which there are hundreds of thousands or millions or tens of hundreds of millions of buying decisions per year – are enormously different from vertical or niche markets, in which there are one or two or dozens or hundreds or thousands of buying decisions per year. Valuation methodologies will vary along this continuum, as one goes from abundant market data to sparse or non-existent market data.

How does one avoid sounding like a piker when speaking with those CEOs above, who may really want to get a grasp on this? How do we convince them to take software valuation seriously when there are few rules; when the answer appears to move around a lot; when the complexity is so high that we must start every answer with, "It depends...."?

I think that we can begin by studying what our panel has had to say and then translating their perspectives into our own situations, tailoring their advice to the multitude of software licensing scenarios that you and I deal with every day.

Bill Elkington is senior director, intellectual property management, Rockwell Collins Inc, and vice president of membership and member engagement on the LES USA and Canada board of trustees vice president

About the Licensing Executives Society (USA and Canada), Inc

Established in 1965, the Licensing Executives Society (USA and Canada), Inc (LES) is a professional society comprised of over 4,500 members engaged in the transfer, use, development and marketing of technology and intellectual property. The LES membership includes a wide range of professionals, including business executives, lawyers, licensing consultants, engineers, academicians, scientists and government officials. Many large corporations, professional firms and universities comprise the LES membership. The Licensing Executives Society (USA and Canada), Inc is a member society of the Licensing Executives Society International, Inc (LESI), with a worldwide membership of more than 10,000 members in 32 national societies, representing over 90 countries. For more information on LES, see www.lesusacanada.org.