MINT~RC

MANAGEMENT OF INNOVATION
AND NEW TECHNOLOGY
RESEARCH CENTRE

SOFTWARE PRICING STRUCTURES IN ELECTRONIC COMMERCE: 3 DIFFERENT CASES

by

Nick Bontis, Ph.D.
and
Honsan Chung

Management of Innovation and New Technology
Research Centre

WORKING PAPER NO. 94
2000

McMaster University
MICHAEL G. DEGROOTE
SCHOOL OF BUSINESS

SOFTWARE PRICING STRUCTURES IN ELECTRONIC COMMERCE: 3 DIFFERENT CASES

by

Nick Bontis, Ph.D.
and
Honsan Chung

Management of Innovation and New Technology
Research Centre

WORKING PAPER NO. 94
2000

SOFTWARE PRICING STRUCTURES IN ELECTRONIC COMMERCE: 3 DIFFERENT CASES

by

DR. NICK BONTIS, PH.D.
ASSISTANT PROFESSOR, STRATEGIC MANAGEMENT
DeGroote School of Business, McMaster University
1280 Main Street West, MGD #207
Hamilton, Ontario Canada L8S 4M4
Tel: (905) 525-9140 x23918 Fax: (905) 521-8995
nbontis@mcmaster.ca

and

HONSAN CHUNG
RESEARCH ASSOCIATE
DeGroote School of Business, McMaster University
1280 Main Street West, MGD #207
Hamilton, Ontario Canada L8S 4M4
honsan@home.com

This article is targeted for submission as an oral presentation at the
1st World Congress on the Management of E-Commerce, Hamilton, Canada.
Track I. ROI for Electronic Commerce/Valuing an E-commerce Businesses

SOFTWARE PRICING STRUCTURES IN ELECTRONIC COMMERCE: 3 DIFFERENT CASES

Abstract

Software is the intellectual capital output of the codified knowledge of a programming team. The development cost is high, but the variable cost of sale is substantially lower (negligible) than for hard goods. Unfortunately, there does not exist a valid or reliable measure to value software. Software is further complicated by architectures. Client-server architectures and combined software-service solutions bring new factors to the pricing decision. With high development costs and low variable costs, one must ponder the methodologies and assumptions used to price software. This paper will examine three cases detailing the rationale behind the pricing decision made and their effectiveness in achieving the business goals of the product (and service). The three cases were chosen with an interest in diverse architectures, usage, and vendor business objectives. Architecture can change many factors such as the dynamics of a market and the cost of ownership to the buyer. Usage stems from both buyer business processes and architecture. Usage is one of the typical metrics of value. However, as the cases are investigated it is shown that in some cases other variables such as flexibility are more important generators of value. To the vendor, the most important metric is the ability to achieve the business goals of the product. A product may be produced as a test platform for future products, thereby lowering the financial performance objectives and increasing the volume and degree of usage as the objective. The business objectives of an application are determined by business strategy that takes into account factors internal and external to the vendor. The negligible variable cost and varying value (per buyer) provides a great deal of flexibility to vendors. The trend has been to align pricing to the activities that buyers realize value from. However, new architectures change the nature of where value is realized and how service becomes part of the equation. There does not exist a perfect generic pricing model. Vendors must understand the value they provide to their customers and create a price structure that aligns pricing with value realization, but more importantly facilitates their business objectives of the product (and service).

Dr. Nick Bontis Bio

Dr. Bontis is Assistant Professor of Strategic Management at the DeGroote Business School, McMaster University. He is also Director of the Institute for Intellectual Capital Research, a consulting firm and research think tank that specializes in conducting knowledge management audits, intellectual capital valuations and organizational learning assessments for clients. Professor Bontis completed his doctoral education at the Ivey Business School, University of Western Ontario. He has won international acclaim for many of his academic research papers, book chapters and business management cases. He is a world renowned speaker and leading authority on explaining firm performance heterogeneity using knowledge-based perspectives.

Honsan Chung Bio

Mr. Chung graduated summa cum laude from the Bachelor of Electrical Engineering and Management program at McMaster University in 1999. His current research interest lies in the business of technology and was reflected in his thesis "Real Options Valuation of Strategic Investments in Uncertain Environments such as the Internet". Mr. Chung also holds three pending patents. Currently, he is an e-Business Strategy Consultant at CBIZ e-Solutions where he is applying his synergistic foundation in technology and business. Other areas of interest include strategic decision tools for hyper competitive industries and Internet business strategy.

INTRODUCTION

The costs associated with the physical characteristics of software are equal to little more than the price of the medium it is delivered upon (e.g., floppy diskettes, CDs, Internet). Hence, the traditional pricing concepts of margins and mark-ups do not apply to software. The software industry is expected to generate $220.6 billion in annual revenue by the year 2002 (IDC, 1999). Software is a critical component of the new information economy. As with all knowledge assets, the valuation of software should be financially optimized. This prompts one to ask how such an intangible asset should be priced?

One may be inclined to approach the issue from a rigorous financial perspective. The millions of dollars spent to create programming code are sunken development costs that must be recovered. In the software industry product development times are short, leaving the cost-recovering first-mover position a short-lived advantage that is rarely sustainable. Other cost dynamics such as imbedding of third party software complicate the financial model further.

Since software development is an output of a programmer's intellectual capital, a more qualitative approach must be considered for valuation purposes. Intellectual capital theorists have argued that the validity and reliability of hard measures for knowledge asset valuation are difficult to develop (Bontis, 1996, 1998; Bontis, Dragonetti, Jacobsen and Roos, 1999; Stewart, 1997; Sveiby, 1997). Bontis (1999) argues that intellectual capital and more specifically, the pricing of software, must therefore take a more subjective approach. Since there is no generalizable formula for the valuation of intellectual capital, there is no generalizable model for the pricing of software.

Mainframe applications, distributed applications and personal applications differ greatly in their usage, importance, and architecture. So, pricing must take into account all of these factors along with an understanding of the value proposition that each software provides to the buyer. Moreover, vendors must align pricing with their goals for the product. The work by the Open User Recommended Solutions Consortium (OURS) task force in 1993 produced a white paper that articulated a set of licensing principles. One of the most important recommendations is that a vendor must recognize that perceived value of software is user specific. However, the white paper places emphasis on pricing on usage.

Due to the complex nature involved in determining pricing models for software, a case-based investigation is used. This qualitative approach provides a richer examination of the alignment between the "value realization" of the consumer and the "business goals" of the vendor.

The key findings of the cases reviewed in this study support the proposition that software pricing is a complex and subjective process. The key determinant of alignment between the vendor and the user is the nature of the value in the software to the buyer. This value proposition may range from increased cost reduction to increased operational flexibility. Successful software vendors priced along these determinants.

REVIEW OF TRADITIONAL PRICING MODELS

Mainframe Processing

Computers have progressed from mainframes and terminals to distributed client-server architectures. Likewise, software has evolved. Banks and large corporations were some of the first customers of software. The software was run on mainframes and priced by calculating the *processing power* of the mainframes as a function of their central processing units (CPUs). A customer with a faster CPU would pay more for the same software. The underlying assumption was that the software was used continually and hence the faster customer would be "using" the software more.

This pricing model makes sense for banks that continually process batches of transactions. However, some programs are not always active or as heavily used among different users. Although traditional pricing models correctly assume that value is realized when the software is processing data, they incorrectly assume that the software is continually run by all users at the same capacity level. Furthermore, a migration via investment in a faster CPU does not create any new value to the user, however the user will be charged a higher license fee. For many years, the mainframe market considered this pricing model to be a fair metric for heavily used software. Unfortunately, many customers felt penalized for owning higher-powered mainframes.

Software pricing based on processing usage did not consider both the needs of the vendor and the user. Vendors had a locked-in market to which they were selling applications vital to the buyers. The goal of the vendor was to maximize both hardware and software revenue which this pricing model allowed. Unfortunately, mainframe customers purchased proprietary platforms with high switching costs and were in fact held hostage by the vendors. Furthermore, innovations in software and increased efficiencies in hardware did not lower costs for users which thwarted proactive developments for operational improvement. To counter these limitations, a licensing model was introduced.

User Licenses and User Classes

As software architectures evolved, pricing models moved towards *named user licenses* and *concurrent user licenses*. Named user licenses are licenses purchased and assigned to specific end-users. This price structure typically does not apply usage limits to end-users. They only specify that licenses are specific to individuals. There is no processing consideration as is the case with the mainframe processing model. The concurrent user licensing structure provides a specific number of licenses that can be shared amongst a group of users. However, the number of concurrent users is fixed. For example, an office with 15 licenses for a graphics application and 20 PCs can simultaneously run the program on up to only 15 of any of the 20 PCs.

This pricing structure has proven popular for software other than database applications as well. In today's business environment stand alone applications and server-based applications are often licensed by this method which is also called *floating licenses*. Companies that offer both structures sometimes price a concurrent or floating license double that of a named user license.

The downside to the floating license structure is that it requires a management system. When end users access a particular software, their PC requests a license from the management system. If the limit on concurrent users has not been reached, the user will be granted a license and the program will be functional. Customers must predict the number of simultaneous users of the application in order to properly size the licensing agreement. This price structure charges customers based upon their peak user predictions and not their average user numbers. Hence, a discrepancy between pricing and value realization may also occur in this pricing model.

In this case, a software vendor is looking for both revenue maximization and market share. Vendors will often give substantial discounts if a corporate site has a majority of computers (i.e., over 65% of PCs) licensed for a given application. This sacrifices revenue, but ensures a market share position via interoperability requirements within a site. The floating license allows access to all computers within a corporate site. Hence the trend to charge a premium over named licenses.

The floating user license pricing improves vendor-user alignment over the mainframe model because both vendors and users can – with investment in special management systems – optimize their positions. To address the capacity issue, some vendors also offer different classes of users such as standard users and light users. The standard user is given full access to all features and the light user is given a restricted set of features. This allows for users of varying needs to be addressed in separate classes.

Although the licensing model improves over some of the limitations of the mainframe model there is the dependence upon peak traffic demands rather than discrete user needs for availability.

CASE A: GRAPHON

The Product and Value Proposition

GraphOn produces software for the thin client concept. Their portfolio of products consists of a server (Global Host) and three clients (Go Global, Go Joe, and Go Between). The server allows clients (on Windows and Java platforms) to access applications on UNIX platforms. An example of this would be a Windows client accessing WordPerfect on a UNIX server (WordPerfect for Linux). In this scenario, the Windows station does not require WordPerfect to be loaded on the local station. Furthermore, the Windows client can access WordPerfect whenever it wants. The value proposition includes both functionality and ease of access. However, the key selling point of the thin-client architecture is the reduction in maintenance cost and complexity at the client side. Overall, it is a reduction in total cost of ownership for the buyer and a flexible platform for applications.

The Market

Thin client software has three markets, which include: i) corporate, ii) original equipment manufacturer, and iii) application/internet service provider. The corporate market is the main target of the thin client concept. Corporate customers see value in the reduction of complexity and in the total cost of ownership (TCO).

With the expansion of the corporate thin client market came a demand from the original equipment manufacturer (OEM) market of vendors creating products for the thin client environment. Currently, the OEM market is large with some of the largest software and hardware vendors as buyers (e.g., SUN, Hewlett-Packard, IBM, etc.).

The third market for thin client software is the application service provider market (ASP). Application service providers are best described as service companies that host and rent out the use of applications. This market is currently small, but it is expected to be the highest growth market for thin client applications forecasted at $2 billion in revenue by 2003 (IDC, 1999). The ASP Industry Consortium membership includes many of the most important software vendors, hardware vendors, service providers and implementers.

GraphOn's main competitor is SCO with their Tarantella product. The Tarantella product does not require client software, only Java capable browsers (similar to the Go Joe product from GraphOn). In terms of the IT market there is competition with the traditional fat client architecture and traditional terminal computers.

GraphOn Price Structure

Though their products will change the way applications will be priced and used, the price structure for the GraphOn portfolio is not revolutionary. Clients are free and servers are priced upon the number of user licenses. There are three tiers for the licenses: 1-10, 11-99, and 100-249 user licenses. There is no limit on the number of applications loaded on the platform. There are two types of licenses: unique-named-users and concurrent users (priced at a premium over named-users).

The support fees are also based upon the number of licenses. These fees are approximately 15% of the revenue from each account. Site licenses are available, but these have only been used for large customers.

Findings

GraphOn is faced with one direct competitor. Hence, their goal (as a platform) is increasing their user base. The portfolio is a substitute product for the traditional fat clients that incur high client-management overhead. The challenge for GraphOn is not their competitor, but convincing buyers to change to their architecture. As an upper bound, the software must be priced at a point where total cost of ownership benefits buyers by allowing for a short payback period (including implementation costs).

The two types of user licenses fit the nature of the two types of buyers. Unique-named user licenses are applicable to corporate buyers that have staff that require frequent access to UNIX applications. In these cases the staff will use the application in daily work for specific applications (regular ongoing usage).

In the ASP scenario, the ASP will be serving out the software to several different client companies with several different users in each. The number of users at one time will vary, but the objective of the ASP is to achieve full capacity usage of the licenses. It should be noted that the concurrent user licenses are not limited to geographic sites, so license capacity usage is possible.

The simple licensing structure allows buyers to quickly evaluate the benefits of the architecture. Value realization is ongoing. Once the architecture is implemented there is an immediate reduction in complexity and operation cost. With this in mind, it can be agreed that the concurrent user license aligns value realization with pricing. Volume discounts (volume tiers) further emphasize the reduction in total cost of ownership, one of the main benefits of the thin client concept. GraphOn has been successful in the corporate market and OEM market. Sales are expected to grow in the ASP market where new pricing options are currently being investigated. Data thus far shows that the price structure was well received by customers.

CASE B: NORTEL

The Product

Advanced Intelligent Network (AIN) software provides telecommunications service providers with a software platform for "value added" telephone services such as *advanced call forwarding* and *automatic call return*. AIN uses a client-server type architecture that is reflected in the hardware.

The traditional mindset of telecom managers has been to equate cash revenues to value. Hence, value is realized when services generate revenue. In the competitive telecom market adaptability also has value. New AIN functionality (new software) increases a telecom operators' potential to generate cash (value).

The Market

AIN software is vendor specific in the sense that only Nortel software can be loaded onto a Nortel hardware platform. However, the interfaces between hardware in the network are open standards. The buyer base is essentially fixed.

The market for AIN has been the large cash-rich Regional Bell Operating Companies (RBOC). Smaller Competitive Local Exchange Carriers (CLEC) have not shown interest due to the high up-front investment and their difference in strategic focus. Currently, software sales are upgrades to new AIN standards allowing for more advanced value-added services. Upgrades are typically released once a year.

Switching costs are high, however it should be noted that vendors sell a portfolio of products and hence customer relationships are important. Upgrades are more important than software proliferation in this particular case. RBOCs consider their voice networks as their cash cow. AIN provides them with "killer apps" such as telecommuting services that leverage their existing infrastructure investment. AIN is viewed as a strong force in bumping up their first priority: revenue generation.

Nortel Pricing Structures

Nortel has tried several different pricing structures to suit the changing demands of customer environments. Below are four of the most significant pricing structures used:

FULL RELEASE / RIGHT-TO-USE (RTU) BUYOUT: This is the traditional software pricing structure with a one-time license payment per client. Some price scaling may be applied to compensate for the differences in density in service provider networks (i.e., telephone lines per switching station). Buyout refers to the purchase of all the functionality in a software release. The upside for service providers is that costs are capped and revenue has a large upside potential. Return on investment can be maximized in this scenario. However, investment is not correlated with value realization. After the purchase the service provider must still market and deploy the service. Furthermore, as time goes on, the service provider may decide not to deploy certain services due to market changes. Hence, most service providers are moving away from this trend.

UNBUNDLED: Unbundled pricing refers to the break down of a software release into a base and optional components. Related capabilities are grouped as components. This aids service providers in that they only buy the components that they require for their service deployment plans. The origins of this pricing structure can be attributed to customers who requested this flexibility. This structure aids the purchase decision because service providers evaluate AIN software with business cases for each capability. This pricing structure has been well received and is the current pricing structure used at Nortel. Only contentious issue remains with this structure in that the up-front purchase price is still not synchronized with value realization.

RISK-SHARING: The risk share is more of a payment structure than a pricing structure. A price is first negotiated for the components of interest. An initial payment is made to install the software and launch the first service. The remaining payments are smaller and are made as each additional service is deployed. This structure is known as option pricing. This structure allows service providers to match their outlays with their revenue streams. There is also flexibility to deploy the services as the service providers sees fit. Because users value flexibility, this payment structure has been successful. However, risk sharing exposes Nortel to the downside risk that the service provider may not deploy further services on the software. In the downside scenario Nortel still acquires the "option" price (premium) in its first installment. In financial terms Nortel sells a call option structure whose value is reflected in the initial payment on top of the price of the first service.

PAY PER TRANSACTION: In an AIN service the Service Control Point (SCP) functions as a database server. The SCP responds to queries by the Service Switching Point (SSP). These query-response pairs are called transactions. Pay-per-transaction (PPT) was requested by service provider executives in recent years (also by the telecommunications standards body). It allows for software costs to be matched with software usage. However, the concept was killed by service provider finance and accounting departments because this metric did not allow for stable cost estimates. Furthermore, costs did not have a cap, unlike the RTU buyout structure. To add further insight, the development of the billing capability cost several programmer hours of development. This scenario reflects the disposition of the RBOC market.

Nortel Case Findings

The purchase of AIN software requires a large investment by the customer. AIN product definition is standards-driven, resulting in functionality development lagging behind identified needs and wants. These demands are not always fully quantified as the pay-per-transaction (PPT) scenario showed. Unbundling software provided customers with the option to choose the functions that they rallied for in the standards bodies negotiations. The one time buyout trend displays the RBOC preference to cap costs and extract the maximum amount of revenue from their investment. Unfortunately, both of these methods do not align pricing with value realization. In response, the RBOCs have begun to see benefits in matching revenue with investments with the advent of risk sharing. The theme of aligning pricing to buyer value realization is emphasized again in this case. Furthermore, it should be noted that the pricing structure evolved over time.

Flexibility to adapt also has value in the increasingly competitive U.S. telecommunications market. AIN software was designed specifically for fast service deployment. With the option to acquire new functionality (i.e., risk sharing), service providers are given a competitive advantage while reducing downside risk. Unbundling and risk sharing aligns the value realization of adaptability with pricing. This is contrary to the natural instinct to view PPT as a better model of aligning costs with value realization.

For Nortel, the full buyout is ideal. However, the option structure increases the upside revenue potential while making the downside a positive premium on the first service. The business objective of revenue increase and up-sell is made to compromise the nature in which the buyer realizes value in the software.

CASE C: HIRESYSTEMS

The Product/Service and Value Proposition

HireSystems is an application service provider (ASP) that has developed its own software. The company offers large corporate clients a web-based, outsourced solution for résumé-processing. The solution includes both software and service. The service component includes résumé receipt, codification, and processing functions. The software component provides clients with post-processing functions such as web-based interview scheduling of candidates and evaluation.

Outsourcing of résumé processing allows buyers to move scarce HR resources away from labour-intensive processing functions to higher value tasks such as employee development and workforce management. The outsourcing of the database and processing reduces the IT requirements of the buyer, allowing them to focus on their core competencies.

The Market

Résumé processing solutions include pure software solutions (i.e., priced using $/seat metrics) and web-based service providers. Web-based providers price based on metrics such as $/hire, $/employee, and $/résumé processed. Pricing models for web-based résumé processing are relatively new compared to pure software solutions. HireSystems must also compete with résumé posting boards such as Monster Board and résumé processing companies like Resumix.

Price Structure for Express

One of HireSystems' services is called Express. Express pricing incorporates both metric and functionality-based tiers. Clients select from three tiers: Select, Gold, and Platinum. Select has basic functionality, while platinum has full functionality. This tiered structure comes from traditional marketing practice, which assumes that if a buyer is given a choice of three service levels, the buyer will choose the middle package. This has been a common marketing tactic in the U.S. cellular service market. Customers are priced according to their volume and functionality needs. Once a functionality level is selected, the customer estimates résumé volumes expected over a given period. Pricing is done on a monthly basis, with a base amount for database management and software usage and a variable amount based upon the number of résumés processed. Unlike the aforementioned cases, pricing at HireSystems bundles software with database management services on a monthly basis.

Findings

As a young Internet start-up, HireSystems is in "build" mode. With a high fixed and low variable cost structure, their business objective is to gain market share. In terms of price points, Express is predatorily priced lower than competitors with equal functionality. The growth plan for Express is to increase functionality. With three tiers, HireSystems can serve a larger range of customers gaining larger market share. The three-tier structure provides buyers with a sense of choice. However, the intention is to sell the middle package.

The most intriguing feature of this price structure is the monthly charge basis. One might initially consider a price per résumé structure but this results in high costs for large clients. HireSystems chose not to follow this pricing strategy because the variable pricing would deter corporate wide adoption by large clients. Instead, pricing was broken down into three basic functions: software usage, database management and résumé processing. Software usage and database management are priced at a flat rate per month. This is logical because expenses do not increase linearly with increases in these activities. On the other hand, résumé processing costs do increase with volume. Hence, processing is charged on a price per résumé basis. The resulting total cost is a function of both fixed and variable costs. This is by design since volume

estimates are acquired before pricing is negotiated. The resulting costs do align pricing with buyer realization of value since costs are associated with résumés processing volume. Furthermore, the on-going IT savings is also accounted for in the base monthly charge. The flat monthly rate also reduces the downside risk for software cost recovery by HireSystems.

To further expand market share, HireSystems negotiates each deal with flexibility. Some customers are temporarily given higher functionality than they pay for. This helps gain new customers. More importantly, it allows for up-sell to the next level once the customer realizes the value of the premium functions. When HireSystems began in 1997 they only had one level of service (Platinum). Gold and Select clients were given the Platinum service until the lower service levels were created. A portion of the Gold and Select clients upgraded to the Platinum package because they found certain functions essential.

HireSystem's Express pricing structure aligns both client value realization and vendor (service provider) market share objectives. The Application Service Provider market is relatively new so more interesting price structures are expected to be developed in the future.

SUMMARY OF FINDINGS

The pricing for each case was specific to the way in which buyers realized value. Furthermore, customers purchasing the same software may actually pay differently, depending upon their own perception of value. In each case the vendor identified the activity through which buyers realized value. This allowed vendors to align pricing with perceived user benefits.

The following chart reviews and highlights value realization, alignment and pricing strategies of the aforementioned cases:

CASE	BUSINESS OBJECTIVE	VALUE REALIZATION	PRICING	ALIGNMENT
GraphOn	Grow user base	Ongoing operational savings	server based upon user licenses	High
Nortel AIN	Maximize revenue	Flexibility to deploy functions	Option pricing on functions	High relative to user perspective
HireSystems	Gain market share	Ongoing savings and per résumé processing	Monthly base plus per résumé cost	High

In the GraphOn case, value was realized by the ongoing reduction in total cost of ownership. The selection of a traditional server priced upon users was applicable because users realized value in usage, but more importantly operational costs were decreased. In the Nortel case

bundling was used to up-sell buyers on more functionality. However, the case highlights the fact that clients realize value when they deploy services using the functionality specifically purchased. This pricing structure is akin to "option" pricing. Like financial options, option pricing gives the buyer the right to purchase functionality when required at a certain price. In this sense buyers realize value in the flexibility and execution of functionality. It is predicted that the ASP market may also acquire this trend to provide the availability of functions and applications to end-users. Unlike the GraphOn case, housing AIN software does not realize value, but having the flexibility to deploy services does have value.

To optimize their financial investment, software vendors must price their product to achieve certain business objectives. In the GraphOn case, growing the user base was the objective. Pricing the server based upon users while providing clients for free allows for an unlimited client/user base. The floating license allows each client the potential to try the application. As the server usage grows, buyers can buy more user licenses. This is a "try and buy" model.

AIN software is expensive to "try". Telecom operators are also locked into specific vendor platforms. Vendors want to maximize their revenue. The only way to increase revenue is to increase the number of functions sold. Buyers are reluctant to invest large sums of cash, but they are open to the flexibility of paying as they deploy. Instead of lowering the total sale value, option pricing provides buyers flexibility, but at a premium price. This increases the upside for the vendor while leaving the minimizing the downside risk.

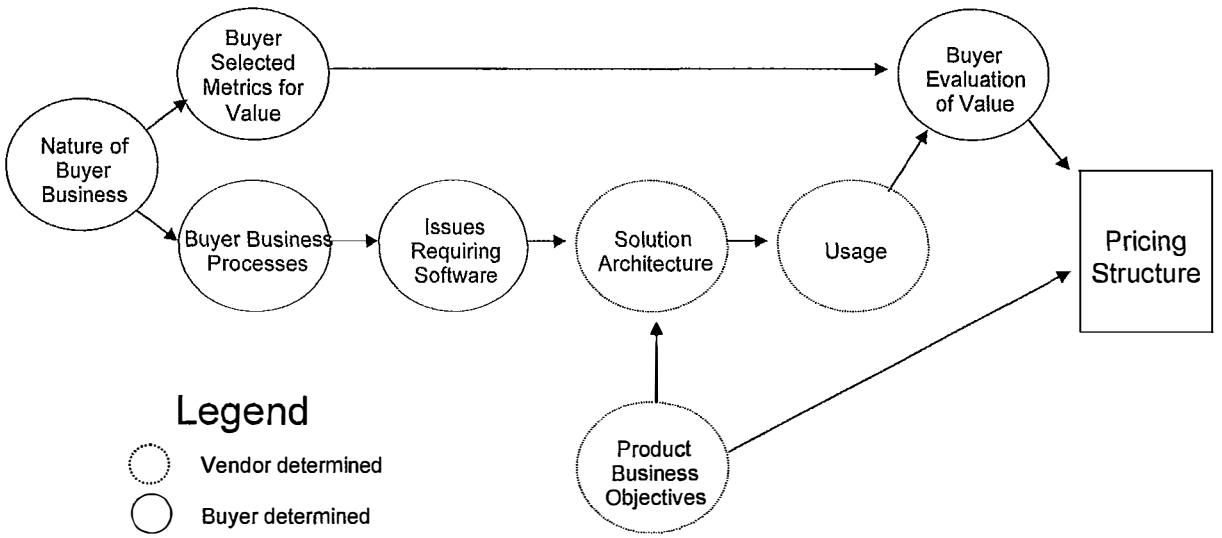
The pricing structure at HireSystems represents the new model for software vendors. The value buyers realize is similar to the traditional benefits of outsourcing. The ongoing increase in efficiency and transactional costs are reduced for the buyer with a constantly improving version of the software. To account for these savings and efficiencies a monthly base mirrors the software benefits while the per résumé charge accounts for the monthly benefits of the processing function. This ASP pricing model is unlike the other cases in that the software is not considered a product, but a service.

Similar to GraphOn, HireSystems' objective is to grow its user base. While in the growth stage of its business cycle, HireSystems must present a relevant pricing proposition to the buyer. The pricing structure designed by HireSystems provides alignment with how and when value is realized by the buyer (i.e., monthly and by volume). This allows the buyer to compare the service relative to the internalization alternative. The tiered structure also provides buyers with a "try and buy" option for higher functionality (a trend in new software).

CONCLUSION

The objective of this paper was to qualitatively describe three different pricing models for software. In doing so, the main proposition revealed in this study is that software pricing must align a buyer's value realization with a vendor's business objectives. This was evident in each case. Though different price structures were utilized for each company, the common feature in each was the effort with which the organization aligned its pricing structure with buyer value realization. Value realization activities ranged from simply having software available to the flexibility of future service deployment.

The second common characteristic revealed in these cases was that the pricing structure designed by the vendors was aligned with its business objectives. Business objectives ranged from user base generation to up selling. The following figure depicts a process flow that outlines a proposed methodology for pricing software:



The process for pricing must consider both sides of the market transaction. However, the dominating feature of the aforementioned pricing process can be found on the buyer-side of the process. It is important to appreciate the variables that buyers will use in measuring value realization. It is these variables upon which the pricing structure should be designed. The source of these variables is the nature of the buyer's business. As with all business understanding the customer is the first step.

Although business objectives for both the vendor and buyer should be considered in any pricing design, it is important to note that successful pricing models use significant weights of emphasis on the importance of value realization on the consumer side.

As the Internet and web-based e-commerce practices continue to proliferate, new innovations in pricing design will surely reveal themselves. However, whatever the innovation, the alignment of both vendor and buyer desires will surely be an important factor in the pricing design.

REFERENCES

Azzara, Carey V. (1999). ***Software Vendor's Guide to Vertical Markets***, International Data Corporation: New York.

Bontis, Nick. (1996). "There's a Price on your Head: Managing Intellectual Capital Strategically", ***Business Quarterly***, Summer, 40-47.

Bontis, Nick. (1998). "Intellectual Capital: An Exploratory Study that Develops Measures and Models", ***Management Decision***, 36, 2, 63-76.

Bontis, Nick. (1999). "Managing Organizational Knowledge by Diagnosing Intellectual Capital: Framing and Advancing the State of the Field", ***International Journal of Technology Management***,18, 5/6/7/8, 433-462.

Bontis, Nick, Dragonetti, N., Jacobsen, K. and G. Roos. (1999) The Knowledge Toolbox: A review of the tools available to measure and manage intangible resources, ***European Management Journal***, 17, 4, 391-402.

McCarty, Meredith (1999). ***Worldwide ASP Review and Forecast***, International Data Corporation: New York.

Stewart, Thomas A. (1997). ***Intellectual Capital: The New Wealth of Organizations***. Doubleday/Currency: New York.

Sveiby, K.E. (1997). ***The New Organizational Wealth: Managing and Measuring Knowledge-Based Assets***. Berrett-Koehler: New York.