Value-Based Pricing Strategies for Enterprise Software

Enterprise technology has entered all the facets of business today. Interestingly, technology sector makes up only a tiny portion of the global GDP. But while IT accounts for 1.5% of the overall global expenditure, the impact that IT has on business performance is enormous. Developers of enterprise software provide immense scope for establishing their pricing strategies based on value the way customers do. Software industry giants traditionally develop their pricing strategies by stressing unduly over the cost-related criteria, at the expense of focusing on the value of software to its customers. These cost-based pricing strategies generate short-term value for vendors, while the value-based pricing strategies generate long-term value for both customers and vendors. The transition to value-based pricing also supports technology shifts that directly impact the design and delivery of enterprise software. The author in his article discusses the merits of value-based pricing strategies in the context of enterprise software. Read on…

Introduction

Leaders in the software industry have traditionally developed their pricing strategies by over-emphasising cost-related criteria, at the expense of focusing on the value of the software to the customer. Cost-based pricing strategies generate short-term value to the vendor. Conversely, value-based pricing strategies are focused on creating long-term value for the customer as also for the software vendor. Today, enterprise technology has become deeply infused within every facet of business. However, it is interesting to note that the technology sector makes up only a tiny portion of global GDP, *i.e.*, a little less than $1 trillion out of the $56-trillion global economy. But, while information technology accounts for just 1.5% of the overall global spending, the impact that IT can have on business performance and value is enormous. Developers of enterprise software provide immense scope for establishing their pricing strategies based on value the way customers do.

The transition to value-based pricing is also because of a considerable level of technology shifts that are directly impacting the design and delivery of enterprise software. These include the service-oriented architecture, component software, dynamic configuration and the provision of enterprise software over the web on a common application platform.

CA. Debadatta Banerjee

(The author is a member of the Institute who may be contacted at debdatta.banerjee@oracle.com.).

In some software market segments like the server operating system and software development tools, these transitions are faster than the others. Customer Relationship Management (CRM) and sales force automation are seeing the shift lately. The back office legacy systems are estimated to transition over the next few years.

Some Traditional Cost-Based Pricing Techniques and Their Drawbacks

1. **User-based pricing:** The charge is based on the number of users that utilise a collection of software features over a given period of time. It attempts to assign costs to a particular number of users or workstations.
2. **Usage-based pricing:** Customers pay only for what they actually use on a transaction basis. This model is also known as *pay-as-you-go pricing* or network-based pricing model. This model charges for outsourced services by transaction, time in use, peak period or some other subscription metric. The application is delivered over the web. In most cases, users pay a minimal set up charge and for service and support.
3. **Tier-wise pricing:** Tiered-wise pricing attempts to package software benefits according to requirements of the users and their willingness to pay. This approach to pricing is an attempt to link software product costs to perceived customer value. For example, Adobe introduced tiered prices for its Acrobat products based on how customers use them, customers who create PDF documents, add comments and sign them and will pay more than customers who only create PDF documents.
4. **Flat pricing:** Users pay a fixed price for unlimited use of the software product. This approach enables customers to more easily predict what they will pay for the use of the software. The fixed price is usually restricted to a particular user or workstation. Many consumer software offerings are priced in this manner and some level of online support is typically built-in for a predetermined period of time. The primary drawback to this method is the lack of flexibility in customising a price for each customer based on the value the customer requires.

Revaluating Pricing Objectives

The objectives of the pricing process are a direct result of a company's overall strategy. A company may pursue a growth strategy of rapidly increasing market penetration and market share. This will require at least in the short run, the adoption of a different pricing strategy than the pursuit of a strategy aimed at increasing profits over time. Pricing objectives are bound to vary by type of software and over time, even within a company and business unit. Although the objective of the pricing process is to determine a pricing strategy which will be a basis for profitable decisions in the medium and long term, pricing strategies are always context-specific and thus bound to change.

Software Pricing in Current Uncertain Environment

Let us briefly observe the fluidic nature of the software industry, the shift of bargain power to customers and the uncertainties surrounding the pricing of enterprise software products and services. Oracle President Mark Hurd, a few months back, commented on the current scenario of traditional enterprise software products: *Think about it: many enterprise apps are 20 years old, and the underlying systems that run them are of about the same vintage. That means they were deployed before the Internet became widely used, before browsers, before smart phones, before social media, before the Web, before widespread BI and analytics, and before tablets. Yet those dinosaurs form the core backbone of many of today's businesses, and they require enormous funding to keep them fed and watered — and that's why there's nothing left over for innovation.*

- In the current environment, the COO has to deal with shareholders perception of IT spending and budget constraints. Software pricing is under pressure from tight IT budgets and customers perception that software is overpriced. Increasing customer focus on business process values are generating resilience to pay upfront license fees for software that supports low value processes or unused components of an integrated business suite.
- In recent times, some software product giants like IBM and Oracle have seen declines in license revenue and thus increased dependence on maintenance or support revenue. These companies are more profitable by not charging upfront heavy license fees but generating a large share from maintenance and support. This phenomenon has given rise to more flexible

pricing techniques like:

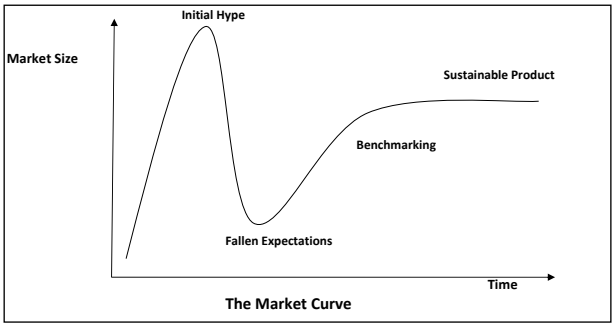
a) term license fees (fixed-term royalty payment basis)
b) software as a service (subscribe to usage of software over the internet)
c) commercial open source software (no upfront or download charges, but charge customer for maintenance or upgrade).

- Large software vendors are refining their pricing strategy based on the industry layer to which a product or a service belongs. At the basic infrastructure level, commercial open sourcing is preferred by the customers, whereas for web-based applications layer, long-term licensing is preferred. Also, service-oriented architecture where the algorithm of software programming is being declared rather than hard-coded, paves the way for future customisation and unbundling of software in an E-Business Suite (EBS).

Market Curve: PLC of New Software

In addition to the general factors like competition, customer preferences and available complimentary products, some specific factors that impact the product life cycle of a new software product are discussed below:

- **Economic factors:** Economic growth dictates the amount of finances that the industry at large is earning and development indicates the volume of money that is being invested into channels of long-term IT infrastructure. Among all economic factors, development is the most important one, as a business has to cater to the demands of an economically-dynamic environment.
- **Technological factors:** The frequency and speed at which old technologies get outdated and new technologies grab the market share is astounding and unpredictable. Some classic examples in the mobile telephony space are Blackberry, Nokia and Motorola, which failed to anticipate the scale of change in customer preferences and evade their natural death. These eventually got acquired by the software companies looking to expand their product range in the telecommunication vertical.
- **Contribution to business value:** Every software that is launched in the market looks to either replace an existing player or create its own space by demonstrating a value. The value that it eventually delivers to the whole business process is critical for the existence of that product or service. Finacle is a core banking software package developed by Infosys Ltd., designed to address retail banking, wealth management, CRM, and treasury requirements of Banks. Finacle is used in 168 banks across 81 countries, since it's initial soft release in 1999 and still going strong in the banking vertical, despite fierce competition from other giant players like I-Flex.
- **Enterprise development costs:** Total development cost to the enterprise is a function of the cost that the enterprise incurs in developing and maintaining the software code itself, comparing the total cost of internal development to sourcing it. A product development activity in the lean market eases the pricing pressure on the new product.



Market curve graphically depicts how the value of enterprise software changes and undergoes customisation since its initial launch. In the coming days, as mobility, analytics, social media and big data get interconnected, the technology will get more complex at the infrastructure level, while the consumption will get simpler at the end user of enterprise software. This volatility will pose more challenge to the COO of software companies, in launching a sustainable product or service while meeting the fast change technology and customer business needs.

Difficulties in Measuring Customer Value

In microeconomic terms, customer value is viewed primarily in two ways:

A. Customer value is the difference between perceived benefits and sacrifices. In microeconomic terms, customer value is seen here as the difference between the consumer's willingness to pay and the actual price paid, which is equal to the *consumer surplus,* the excess value retained by the consumer.
B. A second line of thought defines customer value in a broader way. It defines customer value as

the maximum amount a customer would pay to obtain a given software, *i.e.*, the price that would leave the customer indifferent between the purchase and foregoing the purchase. Customer value in this sense is equal to the microeconomic concept of a customer's *reservation price* and the usage value of goods.

However, measuring customer value for pricing has multiple difficulties. Some foundational concepts of enterprise software value and difficulties of their practical application are discussed here:

1. **Perceived value:** A software product that has a better graphical display may be more pleasing to the eye, but has no value in the traditional economic sense, *e.g.*, a buyer may feel safer buying software from a well known player like IBM if there is a higher perception of reliability, trust and commitment to the market, but it is hard to quantify objectively in product value terms. One technique that is often used to measure the perceived value is conjoint analysis or trade-off analysis. This technique enables managers to compute the consumers' utility functions for individual variables and to understand how they are combined, traded off and individually valued at different levels of price.
2. **Economic Value to Customer (EVC):** EVC is the maximum amount a customer would be willing to pay for the software, assuming he is fully informed about the functionality and competitive offerings. It is analogous to the reservation price. EVC measures the life-cycle economic costs and benefits to the user of one product when compared with a reference product. Software are evaluated on purchase price, installation costs, maintenance costs, operating costs, disposal costs and benefits that can be measured over the life cycle.
3. **Price Sensitivity Measurement (PSM):** PSM models are useful for estimating market demand and for calculating the proportion of buyers that would buy the software within a specific price range. PSM determines the limits of buyer resistance over a range of prices that relate to the software value perceptions. PSM is very useful for pricing alternate software configurations in the early stages of development and throughout the development cycle.

Steps in Determining Launch Price

The first tranche of change has started to indicate the long-term trend that aligns the development cost of software and the business value that the software create. In the days to come, software pricing will become increasingly based on the software vendors' ability to provide differentiating value for the customer.

We will now discuss a few simple steps which will help the decision-makers to gain customer insight on how the software affects her/his business.

1. **Identifying the cost of competitive product and process that consumer views as best alternative:** Any software, process and activity that the customer could alternatively use can serve as reference product. Several software and activities will be able to perform at least part of the functions examined; the economic value of a given software will have to be calculated against at least the principal two and three best alternatives.
2. **Segmenting the market:** The most adept vendors at implementing value-based pricing decisions know that there is no other way of gaining insight into sources of customer value than through observation and intense field research into the customer habits and requirements. Microsoft, for example, is known for handing out beta-versions of its latest enterprise software products to particularly knowledgeable companies and customer segments. This form of free customer feedback is used to determine which features add most value and to gain a deep understanding on how different customer segments use and value the product.
3. **Exploring product differentiation and unique selling proposition:** Software products and services can create value for customers in a variety of different ways: reliability, performance, ease of use, longevity, life-cycle costs, user and environmental safety, superior aesthetics, prestige and so on. The notion of these

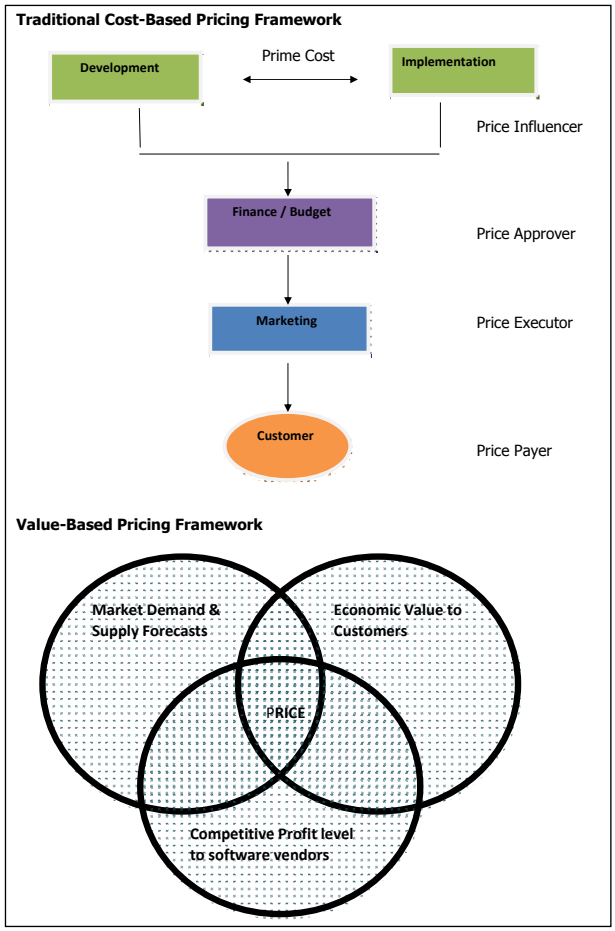
A company may pursue a growth strategy of rapidly increasing market penetration and market share. This will require at least in the short run, the adoption of a different pricing strategy than the pursuit of a strategy aimed at increasing profits over time. Pricing objectives are bound to vary by type of software and over time, even within a company and business unit

differentiating factors is extremely closely related to the concept of competitive advantage. It is important to bear in mind that the customer, not the software development company, is the judge deciding on whether the differentiating factors are actually relevant to better satisfy his needs and future capabilities.

4. **Determining value to customer:** Once the tangible sources of differentiation have been identified, monetary values are assigned to these factors for each identified segment of the market. Expert sales personnel know how to quantify reduced failure rates, start-up costs and life cycle costs in monetary terms, to demonstrate the value of a certain product to actual and potential customers. By presenting options such as (a) a lower price and lesser technical support and (b) a higher price coupled with support and guarantees, software vendors are able to quantify the value of specific product and service attributes for a group of customers.
5. A simplistic approach to value-based price is to sum the reference value and the differentiation value to determine the total economic value. As the price of reference software and the value of differentiating attributes are likely to vary across customer categories, the result of this process is not likely to be one monetary value for any given software, but rather a *value pool* reflecting the fact that different categories of customers will assign different values to the software.
6. The last step is to use the value pool to estimate future sales at specific price points. Once the value pool and economic value profile of a market segment has been determined, sales estimates for different price points can be obtained. For each price point, sales can be expected to comprise a significant share of all market segments which value the software higher than the specific price.

Comparative Framework of Cost-Based vs. Value-Based Pricing

The first flow chart shows the traditional cost-based pricing mechanism. Whereas the subsequent Venn diagram shows the value based pricing technique considering all the three factors, namely, economy, customer and software-selling firm, having an equal influence on the pricing decisions. Firms that invest in a strategic pricing centre can make better decisions throughout the development process, by understanding how customers value product alternatives and arrive at prices that they are willing to pay.



Business Implications in Shifting to Value-Based Pricing

A strategic shift to value-based pricing would bring noticeable change to some key aspects of business operation. Most significantly,

A. **Revenue waterfall:** As the software industry moves to price its products and services in more innovative ways, the business houses witness a *stochastic* shift in revenue recognition; upfront revenue from licensing make way to ratable revenue over a period of time. Any software which the customer accesses not by licensing but

Every software that is launched in the market looks to either replace an existing player or create its own space by demonstrating a value. The value that it eventually delivers to the whole business process is critical for the existence of that product or service

through either a vendor or any other third party is known as a service and not as a product. The revenue for service has to be recognised over the period of delivery and cannot be recognised as upfront revenue from licensing.

In these circumstances, the software seller might choose to recognise the upfront license fees as revenue and treat the maintenance revenue separately to be recognised over the period of support. The software seller also has the option to upfront license the software, with a promise for free upgrade for the next 1-2 years on the same platform. In this case, the entire revenue has to be equally prorated over 1-2 years term from the date of sale. Some bookings fall to revenue whereas some of the bookings fall to deferred, the industry expectancy is to provide more price options to the customers and still retain the total revenue over the life of the product or service.

B. **Cash flow from operations:** Value-based pricing models are certain to bring some significant changes to cash generated from the operating activities. As upfront license payments make way for the deferred and more flexible payment methods, the software developing company has to plan multiple of its activities to suit the available cash flows. A large part of the upfront license fees received in the first year of sale was going to fund the R&D activities and sales team upgrades. The cash flows and expenses are now more carefully matched to time the R&D speed and extent and also that the marketing expenses that trigger cash flows.

While some large corporations already have enough reserves to fund their product and service development and upgradation activities, the smaller software vendors resort to discounting their receivables or from venture capital funding or new stock offerings.

C. **R&D:** Over a longer period of time, R&D activities will shift to the continuous improvement model that focuses on smaller and timely releases rather than major infrequent upgrades, followed by a number of bug patches that often expect the customer to move to a different platform. It is slowly but steadily transpiring that *Software as a Service* (SaaS) will reduce the total cost of ownership to the user over the total period of customer relationship.

Economic value to customer is the maximum amount a customer would be willing to pay for the software, assuming he is fully informed about the functionality and competitive offerings. It is analogous to the reservation price. EVC measures the life-cycle economic costs and benefits to the user of one product when compared with a reference product

As the price of reference software and the value of differentiating attributes are likely to vary across customer categories, the result of this process is not likely to be one monetary value for any given software but rather a value pool, reflecting the fact that different categories of customers will assign different values to the software

D. **Sales:** Pricing methodology will also impact the way the sales team gets compensated. The niche applications players might not see any radical change in their cost of sales, but the traditional software vendors will experience a reduction in number of key accounts. While getting a new customer signing large bookings would become difficult, it would also be convenient to rope in a large customer base as entry barriers fall. This will impact the way channel sales managers or senior sales leaders enjoy upfront commissions under the traditional software pricing mechanism.

Value-Based Pricing for Cloud — Few Illustrations

Cloud as an architecture and delivery model focuses on the shared infrastructure, shared applications and subscription payments. Making transition away from an on-premise enterprise software and moving to a *cloud model* means the entire process is much more relationship focused. The main characteristics are:

1. Cloud applications are typically hosted off-site and by a third-party provider.
2. Cloud applications are accessed *via* the internet.
3. Cloud requires mostly basic IT skills to implement. Since users access the hosted applications online, the specifications of service are simplified and there is no lengthy time to implement – unlike many on-premise systems.
4. Inherent to the Cloud delivery model are self-service requesting, near real-time deployment, dynamic and fine-grained scaling.

Value-based pricing models are certain to bring some significant changes to cash generated from the operating activities. As upfront license payments make way for the deferred and more flexible payment methods, the software developing company has to plan a multiple of its activities to suit the available cash flows

Now let us consider some unique techniques for pricing of cloud services:

- **Freemium model:** With freemium model, the core functionality is provided free and establishes a pay-for upgrade path that includes increased functionality. This model is recommended when the value of offering is derived from some sort of collaborative capabilities. With a freemium model, if it is hard to demonstrate the value in moving to the paid version, there is a risk of getting stuck with a large user base and no revenues. The important message here is to be very careful on how to segment the functionality between free and paid software. LinkedIn, a social networking site for business professionals, is a good example of a successful freemium model, generating revenue in three ways—premium subscriptions, corporate solutions and advertising.
- **Limited-feature model:** With this model, the software companies provide a relatively low-cost version of software service, to encourage rapid adoption but limit features to encourage upgrade sales. The price should be set relative to other available solutions, *e.g.*, the first vendor to offer a SaaS solution in a market that has traditionally been served by on-premise solutions can use the established market value as the entry point for that functionality to set it's own pricing. The vendor could also set the price at a discount from the market value and emphasise the benefits of SaaS to the customer (time value and lower upfront costs for deployment, *etc.*). The two US based dominant players in the Cloud space Salesforce.com and Workday.com have consistently done a good job with the feature-limited model in their entry level offering with revenue growth derived from organisations that increases their user bases as well as move up to higher-priced and more featurerich editions.
- **Pay-as-you-grow model:** *Pay-as-you-grow* model is a usage or transaction-based model in which customers only pay for what they use, with no recurring or base fees. This model is dynamic and allows the most flexible pricing for customers. With this model, customers take on very little upfront risk. Pricing is on per-unit basis and is generally higher than fixed or tiered plans.

IBM Senior Vice-President Erich Clementi says: *We fully recognise that SMB (small and medium businesses) customers have different sets of needs and IT requirements than our large enterprise customers. And just to be clear, these are not watered-down, leftover, or recycled enterprise level products. Instead, the offerings feature a portfolio of easy to deploy, rapid-value solutions that are appropriately sized and priced for SMBs.*

However, customers are also increasingly facing a modern business hurdle of hosting their critical applications on the cloud; staggering costs when data-centre breakdowns trigger lost revenue, customer disenchantment, dropped employee productivity and injured brand image.

Conclusion

Large software corporations are rethinking their pricing strategy and portfolio of products or services and carefully choosing the applicability of value-based pricing, stacking up their portfolio with offerings which have more value. The emerging trend is that the basic software will become virtually free-of-cost or be supplied by some vendors trying to hold their market share with provision of basic but essential hard coded software. By moving far beyond the traditional approach, engineered systems help CEOs tackle the IT-infrastructure sprawl whose insatiable appetite leaves precious little funding for customer-facing initiatives. The author strongly believes that the software industry is fast pacing towards a model of actual value delivery to customers. Pricing is no more the value that the large corporations place on the enterprise software, but the actual benefit that the customer receives based on the criticality of the business process.

Pricing approaches based on customers' perceptions of value are strategic and long-term in nature, since they are focused on capturing unique value from each market segment through the pricing mechanism. Enterprise software selling companies need to invest to create pricing capital, to ensure the long-term benefits of value-based pricing. ■