



WHITE PAPER

Lucrative Pricing and Packaging Strategies for the Cloud Today, Tomorrow, and Beyond.

Executive Summary

Software pricing and packaging is an art form regardless of whether it's delivered as a service or as physical on-premise software. There is also a lot of science involved. This paper explores the most critical aspects of introducing and managing SaaS applications and presents ISVs with ideas on how to build, execute and manage effective software pricing and packaging strategies for Cloud services.

You will find valuable information to help ensure success, whether your objective is to build a customer base for a new offering, to introduce a Cloud service to an already established on-premise customer or, ultimately, to migrate all of your on-premise business to the Cloud. This paper looks at emerging best practices and real-world ISV successes that will help you feel confident in your ability to tackle software pricing and packaging now and in the future.

A common framework has emerged for how companies are developing successful strategies for pricing software in the Cloud.

Pricing & Packaging: Both Art & Science

Pricing for software and service is very much an art but there's also a lot of science involved. A common framework has emerged for how companies are developing successful strategies for pricing software in the Cloud. As the SaaS model continues to evolve, there are a variety of case studies and best practices we can use to identify the typical components of almost every successful SaaS strategy.

How SaaS is Different

One of the things that software as a service enables ISVs to do is to get away from what's been wrong with traditional on-premise software pricing in the past. In the distributed-computing client-server days on-premise software was sold through a perpetual license with an added maintenance contract. With a perpetual license model, customers are incented to buy a lot of seats up front and then pay an ongoing maintenance fee on top that each year. For many reasons this does not work in every customer scenario.

That being said, most software revenue today is still associated with the perpetual "plus maintenance" model. However, it is important to note that the fastest growing portion of today's software industry is associated with subscription revenues, whether a software as a service or an on-premise solution. With software as a service, you are giving resource access to a customer over the Web as a service. There are some interesting things you can do in terms of the types of metrics and the types of meters that you offer, just by nature of the fact that you or a service provider that you designate is running a single instance of the software application for many customers.

There is a great deal of tracking you can do in that environment that is hard to do in an on-premise environment, especially after the fact when the customer has already installed the software and has an established way of pricing. So that, in and of itself, and the whole subscription model, brings many unique features and attributes to a fast-pricing approach, and also opens a realm of possibilities that really haven't been available in the software world before.

Making a transition away from a physical product and moving to a SaaS model, means the entire process is much more relationship focused.

And as transition is made away from a physical product to a SaaS model, the entire process becomes much more relationship focused. So instead of a physical product, your customers will receive access to software resources that will help complete a business process for them. You, the vendor, are providing the software service and the customer is making an ongoing decision as to whether to pay for it based upon the value they derive. SaaS is much more of an iterative process, which can start off small and continue growing over time.

The high-level description of SaaS as an architecture and delivery model focuses on shared infrastructure, shared applications, and subscription payments.

As an ISV considering the software as a service business model, it's important to understand exactly how SaaS differs from traditional on-premise product and business models. The high-level description of SaaS as an architecture and delivery model focuses on shared infrastructure, shared applications, and subscription payments. In theory, it all sounds rather easy for a traditional ISV to deliver. But in reality, that simplicity can sometimes mask the complexity required to execute SaaS correctly.

Software as a Service Characteristics and Requirements

There are many service characteristics that ISVs have to consider. As you look at the following list, you will see that each scenario carries a new perspective on software creation, delivery, and the product lifecycle, which most traditional providers don't have. There are also important requirements that ISVs considering adding a SaaS product or fully transitioning to SaaS will have to meet.

1. SaaS applications are typically hosted off-site and by a third-party provider.

"In-the-cloud" execution, for most practical purposes, means your software is hosted at an off-site location by a third-party provider.

When software is offered as a service, ISVs or their designee provide users with a Service Level Agreement (SLA) that will guarantee most aspects of service delivery, including technology and customer service.

2. SaaS applications are accessed via the Internet. SaaS applications are accessed through a standards-based, universal network which also requires ISVs to provide N-scalable service.

3. SaaS requires minimal or no IT skills to implement.

Since users access your hosted applications online, the specifications of service are simplified and there is no lengthy time to implement – unlike many on-premise systems. As a result, ISVs are required to provide real time service and, in many cases, are working with non-techie types within the enterprise.

4. Automated provisioning. – Inherent to the SaaS delivery model are self-service requesting, near real-time deployment, and dynamic and fine-grained scaling. It's a transactional environment that requires ISVs to provide dynamic scaling and flexing of capacity.

5. Fine-grained pricing. Most SaaS contracts include fine-grained pricing based upon consumption. ISVs must be able to provide and support usage-based pricing (though some mask this granularity with long-term, fixed price agreements.) ISVs pricing strategies should be based upon on-going delivery of shared value.

6. Shared resources/common versions – Configure vs. customize, with integrations and client-side optimization. ISVs provide a single code base, multi-tenant architecture.

Key Components of SaaS Contracts

There are many interesting and exciting things you can do with SaaS packaging and pricing. What follows is an explanation of some of the key components that should be part of every SaaS project.

A **subscription model** is the first and most basic component. With a subscription model, users pay on a recurring basis to access software as an online service. As long as the customer continues to pay for that access, they get access.

Pricing metrics are another component of SaaS contracts. Pricing is typically user based. With SaaS, there are a variety of user-based metrics you could use because of the metering and tracking capabilities available when you run an instance of an application for the customer, as opposed to the customer running an on-premise application. The most simplistic example would be to measure how many business users are accessing the service. But much more sophisticated scenarios exist. For instance, which features are used most often, or which features are most popular at a particular time of day or particular segment of the market? There are many possibilities. But presently, there is not as much experimentation among SaaS vendors as you might expect.

The **contract term** is also a key component of any SaaS contract. From a term perspective, you could specify month to month or multi-year, but, on average, where things tend to pan out is with an annual contract. And here's why. As a consumer or a business user, do you really want to cut a PO every month? Do you really want to go through the decision-making process on this Cloud application every month? No. Once the decision has been made, most customers would rather just set it and forget it. For trials or evaluation purposes, it might make sense to offer shorter-term contracts. But once customers are ready to commit, chances are they will sign a one-year contract. This is the nature of a subscription approach.

One of the customer challenges (or fears) when it comes to contract terms is that once they become reliant on the application, the ISV will raise the price for the next term. It's a very common scenario. And so to lock in pricing, business customers will often negotiate multi-year contracts. Best practice advice is to take a very customer-friendly approach and

offer term options. With SaaS contracts, it's quite common to see a one-year, a three-year, and even a five-year option. And the per-unit price is reduced as the term scales out in time.

Subscription revenue must be recognized as the service is delivered

Two more important contract components are **billing** and **revenue recognition**, which go hand in hand. Billing, even if it's for a one-year term, is typically up front. So a customer that signs for a one-year term generally pays for that one-year term at the beginning of the service. The revenue, of course, is not recognized at that point in time because subscription revenue must be recognized as the service is delivered, which is part of the revenue-recognition rules around subscriptions and software as a service.

Maintenance revenue terms work the same way. Revenue recognition is ratable over the term of the agreement. Though SaaS implementations are much shorter and on a much smaller scale than an on-premise implementation there may also be professional service start-up charges associated with signing a new customer. The professional services component is recognized on a different schedule than the actual subscription service.

A **Service-Level Agreement (SLA)** is also an important component of SaaS contracts. When software is offered as a service, ISVs or their designee provide users with an SLA that guarantees most aspects of service delivery, including technology and customer service. The customer service guarantees often include availability of support resources and response time on technical support requests. The technology guarantees can include error resolution time guarantees, system response time guarantees, and almost always include system availability or uptime guarantees. If service levels are not met, it can negatively impact revenue for the ISV because money is given back to the customer. SLAs with money-back terms are a standard that started with telecom service providers and ASPs many years ago.

A component that is unique to SaaS contracts is the **dynamic scale up and down**. If you consider the on-premise, traditional license-plus-maintenance model, there really is no way to scale down from an accounting perspective, especially because perpetual license revenue is recognized up front. It is difficult to give licenses back to a software company that's already recognized that revenue. There are sometimes exceptions made for very large and important accounts in specific cases, but it's definitely not part of standard policy.

On the flip side, when subscribing to the software and receiving access to it during a defined period of time, the scaling up and scaling down is a lot easier from a back-end accounting perspective—and it's certainly expected from a customer perspective. When you get to the more utility or actual pay-per-use (not pay-per-user) pricing models, you have to think about using a metered approach that scales up and down dynamically, meaning your customer might have different prices they're paying within a month, or even within a day.

Keep it Simple but Not One Size Fits All

Additional SaaS contract considerations include predictability, setup fees, minimums, and differential pricing.

Predictability. With SaaS, predictability to ISVs means consistent costs (the cost of service per user) and recurring revenue. Predictability to users means pricing plans that provide cost smoothing over time and cost based on consumption. These pricing criteria make it possible to plan for more users, or increased transactions, or increased storage needs, so that they can budget accordingly.

Best practice advice is to build pricing plans with distinct cost-based and value-based components with relatively flat on-boarding costs to users. ISVs should also consider building an array of tiered high-value offerings. Some examples include storage, network optimization, integration, data cleansing/analysis, test and code QA for customizations, advanced billing automation, etc.

A poor economy and shrinking budgets have made it essential that ISVs ensure SaaS customers quickly receive value for their money.

Setup Fees. Traditionally, as a penalty to those users who chose to defect, SaaS providers charged high setup fees. This practice is now shifting. A poor economy and shrinking budgets have made it essential that ISVs ensure SaaS customers quickly receive value for their money.

Best practice advice suggests vendors remove or dramatically constrain their setup fees. Before determining setup fees, ISVs should ask themselves: how much will implementations actually slow down the sales process? What amount is reasonable and how does this charge influence my monthly recurring prices? The goal is to remove the barriers and to quickly get the customer to value.

Minimums. Several SaaS vendors employ usage-based pricing models similar to the usage tiers on cell phone plans. Like cell phone carriers, these ISVs have chosen to decrease their implementation fees in order to lock in higher minimums.

Differential Pricing. Most vendors calibrate their SaaS licenses so that there is a three to four year break-even point versus on-premise licensing investments. Larger, more established vendors typically do not give customers a price break, but, instead, give them more options. The up-front design of differential price plans is based on a variety of things, including brand maturity and premium price, and must be set in the minds of buyers.

For example, consider an established ISV with an established enterprise e-mail product, with several hundred million dollars in annual revenue and a very strong user base that employs differential pricing. The ISV knows the average cost per user to run their product, which includes license, maintenance, support, full-time staff, hardware, utility costs, and so on. Even though users can get a comparable hosted e-mail solution for only pennies per user, this established ISV, with its premium franchise, commands a consistently higher price.

So when they rolled out their hosted e-mail solution, they could employ differential pricing. They used a benchmark price and positioned the new offering at a slight advantage to it – which, in theory, controls price erosion. It is important to note that customers have indicated repeatedly in surveys that they feel the value in multi-tenant architectures is that the provider will pass on the lower costs to the end user. Differential pricing that is too high is seen as out of touch with a cost-plus method or the value of the offering, and buyers may avoid it as not truly new.

It's a balancing act to keep it simple but not one size fits all. Balancing the flexibility as your pricing evolves, while also continuing to meet the needs of your customers as they become more mature and more enterprise oriented. It's important to remember that the roots of software as a service and the Cloud are in simplicity.

At this point, you may be wondering if there are ways to easily address your pricing and packaging strategies for SaaS. The answer is yes. And it all begins with your business objectives.

Pricing and packaging strategies should always be driven by your business objectives.

Pricing & Packaging Begins with Business Objectives

Pricing and packaging strategies should always be driven by your business objectives. In other words, step back and take a look at the big picture. What are you trying to accomplish?

For example, are you trying to grow your customer base for a new entry into the market, or maximize revenue from a mature offering, or trying to differentiate your offering to defend against competitive pressures?

This paper looks at pricing and packaging from three different perspectives, and explores some of the objectives and strategies for each.

1. A new pure-play SaaS startup, really moving out with a SaaS-only offering.
2. An established ISV with an existing on-premise portfolio, attempting to launch a new SaaS application to a new market or customer segment.
3. An established ISV with an existing on-premise portfolio and a strong customer base wishing to augment the existing portfolio with SaaS or even migrate over to SaaS completely.

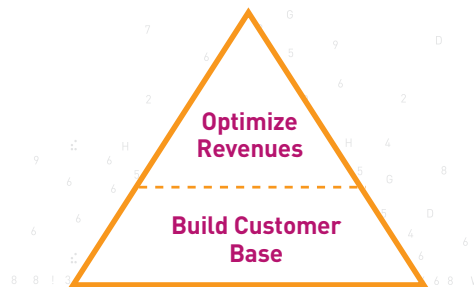
Whatever your objectives, they undoubtedly will influence which pricing and packaging strategies you choose to help meet your financial goals.

The key objectives are to build your customer base and then to optimize revenue.

Kick-Start a New SaaS Offering

In either of the first two scenarios above, the key objectives are to build your customer base, and then optimize revenue.

The first step in establishing your new SaaS product or business is to build a customer base. You need customers for market validation, revenue, product direction, and to stake a claim against potential competitors.



Next, you'll be focused on continually growing your user base and, as the offering matures, you will need to start concentrating on optimizing revenues through the monetization of new features, and then by differentiating your SaaS application through packaging and various licensing models.

Establishing Market Share

The key to establishing market share is in tying the value of your service to your pricing model and balancing that with your business objectives. This may sound obvious, but it is important to emphasize that one of the benefits of SaaS is that you can monetize in so many different ways.

One of the benefits of SaaS is that you can monetize in so many different ways.

When the goal is to quickly gain customer adoption and market share, it needs to be easy for customers to experience the core value of your offering with little or no barrier to entry. The following examples are just a sampling of the various models being used today to build a customer base and establish market share.

Choosing which model works best for your business will depend upon your objectives, your application, and your target market.

The Freemium Model

With the freemium model, you provide your core functionality for free and establish a pay-for- upgrade path that includes increased functionality.

This model is recommended when the value of your offering is derived from some sort of collaborative capabilities. Or, in other words, where there's more value derived by individual users and their interaction as the user base grows. Another scenario where freemium is a good choice is when the technology or the functionality is not well understood by the market. The freemium model is also a good option when there is a substantial increase in value to the user if they upgrade from the free version to the paid version. With a freemium model, if it is hard to demonstrate the value in moving to the paid version, you run the risk of getting stuck with a large user base and no revenues. The important message here is to be very careful how you segment your functionality between free and paid for.

LinkedIn, a social networking site for business professionals, is a good example of a successful freemium model. LinkedIn provides a free framework upon which users can maintain their professional network of present and prior work associates, industry colleagues, and contacts. LinkedIn generates revenue in three ways—premium subscriptions, corporate solutions (such as recruiting services), and advertising. They are operating profitably and 2009 revenues exceeded \$100 million. Estimates for 2010 put them at close to \$200 million—Impressive growth fueled by a rapidly growing user base and a clear differentiation in value derived from the premium subscriptions.

Best practice...suggests you take away the barriers to entry. Set a relatively low price point for a limited feature set and minimize or avoid set-up fees as much as possible.

The Feature-Limited Model

With the feature-limited model, you provide a relatively low-cost version of your software service to encourage rapid adoption but limit features to encourage upgrade sales. The price should be set relative to other available solutions. For example, if you are the first vendor to offer a SaaS solution in a market that has traditionally been served by on-premise solutions, then you can use the established market value as your entry point for that functionality to set your own pricing. You could also set your price at a slight discount from the market value and emphasize the benefits of SaaS to the customer (time to value, lower upfront costs for deployment, less commitment, etc.). If this approach sounds familiar it's because it is one of the models that has been used for on-premise software for many years. However, with SaaS, there are some key differences to consider.

- > A feature-limited model is best suited to SaaS offerings that do not rely on collaboration or community interaction to drive business growth.

- > A feature-limited model is better suited to offerings that have an established market value.
- > This can be the best approach for those providers that are unable to shoulder the costs of delivering a free service in scale.

If you employ this model, best practice (as mentioned when discussing contract components earlier) suggests you take away the barriers to entry. Set a relatively low price point for a limited feature set and minimize or avoid setup fees as much as possible. You will also want to limit term commitments as much as possible; likely nothing longer than one year. Depending upon your market, you may even want to adopt a no-contracts approach to start out. You can then incent customers to move to higher price points, longer term commitments, and the like, based upon how you package and price your capabilities. It might also be beneficial to focus your limited features on solving narrower use cases

Salesforce.com has consistently done a good job with the feature-limited model in their entry-level offering, with revenue growth derived from organizations that increase their user bases, as well as move up to higher-priced, more feature-rich editions.

The Pay-as-you-Grow Model

The pay-as-you-grow model is a usage-based 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 up-front risk. Pricing is on a metered or per-unit basis and is generally higher than fixed or tiered plans.

Real-world examples of services, in which value is closely tied to usage, are Amazon Web Services (AWS) or Skype. The pay-as-you grow or metered use models can be very desirable when trying to grow your customer base. Once your customers become accustomed to your offerings and develop a consistent baseline of usage, there will be opportunities to grow revenues and reduce their potential variability.

These are just three of the many options available with SaaS. And, once again, choosing which model (or models) will work best for your business depends upon your objectives, applications, and target markets. Once you've built a solid and growing customer base, you need to turn your focus to growing your revenue.

The pay as you grow or metered use models can be very desirable when trying to grow your customer base.

Increasing Profitability

After you've developed a solid customer base, it is important to focus on growing revenues—through your existing customer base and by expanding into new markets.

Take, for example, Skype, which is a free service. Beyond the free offering, they have subscription models, as well as usage

models. They've based their packaging and license models upon the different segments in their customer base.

Another example is a network appliance provider that is considering adding services to their existing product portfolio. This provider believes they could maximize their profits with a user-based licensing model, but they recognize that in order to optimize revenue, they'd have to offer subscription-based licensing, as well as fixed-base pricing, in order meet the needs of all customer segments.

Another example is a pure-play SaaS startup, which went through over 50 pricing and packaging changes in the first 18 months following launch. To gain traction and grow the business, they were constantly changing feature sets and pricing approaches. Although this may be an extreme example, this is common feedback about the need to be able to rework pricing and packaging during the early stages of launching a service and on an ongoing basis. Customers expect this agility in the SaaS world where the perception is that it is easier to adjust packaging and licensing more dynamically.

It is important to be flexible in packaging your software services and to offer diverse licensing options. The examples above help illustrate how important it is to not only look at which license models work best for your application and your business but also for the value proposition that your customers fit within.

It is important to be flexible in packaging your software services and also to offer diverse licensing options.

Evolving an Established On-Premise Strategy to Include SaaS

If you are an established ISV with a traditional on-premise software offering that you want to augment with SaaS, you will be referred to as an ISV with a hybrid offering—meaning your portfolio will include part on-premise software applications and part SaaS applications.

A hybrid provider is much like a hybrid vehicle. A hybrid vehicle has a traditional gasoline engine that fuels the car. This engine is like the on-premise portion of your portfolio. Now, adding a small electric motor to the hybrid vehicle makes it much more efficient, just as adding SaaS to your on-premise portfolio can bring efficiency to your overall business.

It is important to be flexible in packaging your software services and also to offer diverse licensing options.

SaaS can complement the investments you've already made in your on-premise portfolio. And it's important to think of this not only from your viewpoint but from the viewpoint of your customers. The investment that your customers have made

in your on-premise offering over the years, and the value they continually receive from using your software, has become ingrained.

They are comfortable using your on-premise offering, so extending the existing value with services allows you to reinvest and increase the value that they receive. This helps to protect your current maintenance revenue and also protects your customer base, and keeps them focused on the value you bring to them. Looking at how SaaS can augment on-premise offerings to solve problems is an important strategy for established on-premise ISVs.

They are comfortable using your on-premise offering, so extending the existing value with services allows you to reinvest and increase the value that they receive.

Hybrid Pricing and Packaging Strategies

Next, we'll explore some of the pricing and packaging strategies for hybrid offerings. But first, let's look at yet another parallel between the hybrid vehicle and a hybrid product offering. We'll call it preferred parking. If you don't yet drive a hybrid vehicle, you probably get a little frustrated when you see one of those signs in the parking lot that reads 'Hybrid Parking Only.' You're thinking "Wait a minute. Why don't I get preferential treatment?" A hybrid software product offering is similar to preferred parking. And here's why.

The message with this parallel is that if you're an ISV with a mature on-premise offering, a strong customer base, and a SaaS offering, you will receive preferential treatment because you'll be able to demonstrate the flexibility of leveraging your on-premise deployment and service-based offerings to meet the needs of most any scenario.

When trying to identify how you can use SaaS to augment your current on-premise portfolio, there are many different ways to think about it. One approach is to offer complementary functionality in your SaaS offering—functionality that fits nicely with what you're already doing on premise.

Another option is to focus on increasing the value of the on premise offering by solving use cases that cannot be solved via an on premise solution. An example of how one CAD ISV is accomplishing this is by adding social community features to their offering. The social interaction and collaboration allows them to more efficiently gain feedback through the supply-chain process, the design process, and the development process.

In addition, it adds value for them and their customers because they are able to take the collaborative intelligence gained from the user community and implement it in the form of features and functionality that solves problems. The community functionality offers customers better visibility into what the next iteration of the product will be and an opportunity to interact during the development process.

This use case could not be solved well using their current on-premise offering.

You could also repackage your current offering to solve different use cases. Say you have an accounting application with a rich set of features. The core users of your application, the ones who actually use the majority of those features, represent only about 20 to 30 percent of the larger organization. The rest use only a small subset of the available features. This scenario is quite common and closely follows what has become the industry standard. But, still, there is value in creating an abbreviated version to extend to the 70 to 80 percent of users within that organization who would prefer it. It may be best to offer it as a SaaS application. There's really no reason to repackage or deliver another on-premise instance of your application when you could carry out a new SaaS-based offering with just a few features to solve that need. It is important to look for these types of opportunities.

Another good example is a graphics design company that sells to the SMB market. Internal IT resources are generally very limited for most small- and medium-sized businesses, so SaaS applications that require no internal IT staff are very attractive. This graphic design company plans to add two new service offerings to their existing portfolio. The first is an external storage capability to provide redundancy to their customers' environments. The second is a light version of their on-premise graphics application, as a service. The SaaS version would essentially provide a markup capability. Beta feedback for the two new SaaS offerings has been very positive.

When selling into your existing customer base, the ones driving the SaaS initiative within your business will have to come to terms with all the will take place.

Easing into It

When selling into your existing customer base, the ones driving the SaaS initiative within your business will have to come to terms with all the changes that will take place. Once they do, it doesn't mean that your customers have, and it doesn't mean that the rest of the organization has either. It's a transition. It is important to be considerate of the need to ease into adding a SaaS application to your portfolio.

Initially, one way to do that is to use license models that fit the current on-premise offering. Perhaps you don't begin with a usage-based model. Maybe, for example, you start by offering your service as an annual subscription and align it to your maintenance renewals so that your internal organization doesn't have to change their processes too much in order for you to get that service in the market. Over time, you can introduce new flexible license models that better fit the service offerings you've brought to market.

The Art & Science Explained

It is very important, as a first step, to figure out how you are going to enter the market from a pricing, packaging, and overall strategy perspective—flexible subscriptions, usage-based models, maybe even prepaid pools, such as a pool of seats. It could also be a pool of usage; whatever that unit of usage may be. You should also be thinking about even more flexible options, such as overages, rollovers, or peak usage, which are popular in the mobile space.

When developing the price point, the average break-even point for the customer, where the cost of a SaaS subscription starts to become greater than the cost of your existing perpetual model, is somewhere around year three to four.

Figuring it all out starts by doing some homework and building a business case. Generally, ISVs begin with their on-premise perpetual license number and then think about things like current maintenance renewal and maintenance attrition, and the existing costs of goods for their perpetual model. Next, they consider assumptions around what their renewal rate might be with a subscription.

When developing the price point, the average break-even point for the customer, where the cost of a SaaS subscription starts to become greater than the cost of your existing perpetual model, is somewhere around year three to four. It doesn't matter what type of software you have, or what type of customer you're selling to, the industry benchmark standard is three to four years.

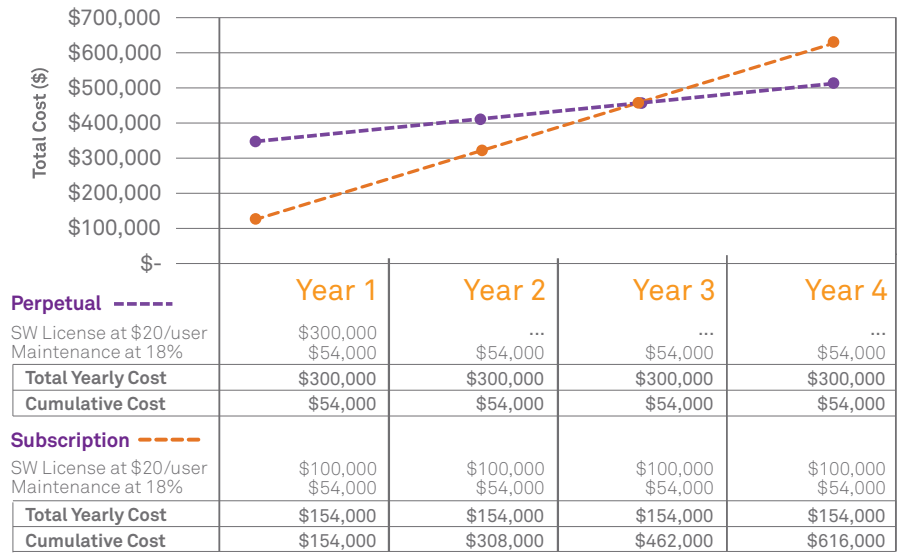
You can then calibrate your SaaS subscription price based upon your own objectives while considering that three-to-four-year break-even point. So, for example, if you want to incent people to move over to the SaaS subscription, you'll push that break-even point out. If you want to keep customers on the perpetual model, which many cautious vendors do, then reel in that break-even point.

The other thing you need to think about as you build this out is whether you want to offer one-year, three-year, and five-year options. And, obviously, there will be some advantage to the customer, from a pricing perspective, if they choose the five-year option versus the one-year option.

These things essentially cover the science part of the pricing and packaging formula. Everything else you consider is art because it will have some element of your unique situation and orientation based upon your objectives, applications, and target markets.

Customer Break-Even Point Perpetual vs. Subscription

- Example for 20,000 users
- Industry benchmark is equal cost to customer in 3-4 years
- The higher the switching costs, or more sticky the application, the closer to 3 years



The Cultural Factor

The cultural factor, which is essentially the hunter/gatherer sales analogy, comes into play quite often with an already established software on-premise vendor when attempting to augment or fully transition the portfolio to software as a service. Everyone is excited. SaaS is going to the high-growth portion of the overall company, but oftentimes those people end up getting very frustrated because they realize that although they have committed to the idea of software as a service within their business, the culture of the sales organization is still oriented around the traditional on-premise business.

The perpetual license models of packaged software companies have spawned a culture oriented around the hunt. Going out and getting the biggest deal, signing the giant end-user license agreements, and then figuring out what to do with the accounts later. This type of culture doesn't really put much of a premium on selling subscription contracts that start off small and maybe grow over time, and they definitely don't put a premium on those if they interfere with their ability to hunt the large deals.

You can come up with all the spreadsheets and build the best business cases, but unless the entire company is truly committed to making SaaS an important part of their business, it probably will not work.

You really have to think about the orientation of your company. The cultural factor cannot be ignored because it impacts how successful you can be. You can come up with all the spreadsheets and build the best business cases, but unless the entire company is truly committed to making SaaS an important part of their business, it probably will not work. You need to foster the gatherer mentality within the organization. Making sure

you live up to your commitments. Helping customers feel so overjoyed that they want to continue to pay over time. Everyone, from the CEO on down, has to be committed to orienting the business around this subscription culture, and the sales culture must be re-oriented around harvesting that money over time.

The cultural factor is something that is difficult to define but it is extremely important. It is the very reason so many software-as-a-service efforts within large traditional software companies have not been as successful as they could be.

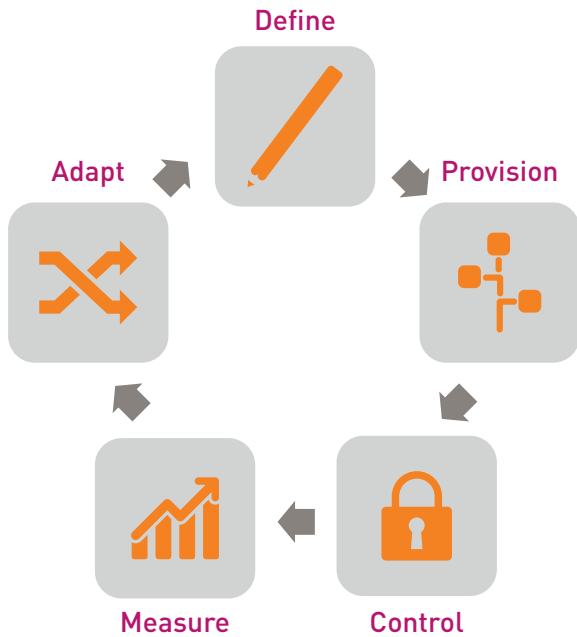
Conclusion

There are there many ways that technology can help mitigate the simplicity and flexibility challenges faced when pricing for the SaaS customer base. From a technology standpoint, the balance between simplicity and flexibility is an important one. To be successful with the SaaS experience you need the following components to effectively deliver on that balance of simplicity with flexibility.

A successful SaaS experience requires the ability to comply with service agreements, versatility in packaging, the agility to quickly address new opportunities with sophisticated licensing models, the ability to gather business intelligence to support usage-based billing, and strong back-office support to deliver an integrated on premise and SaaS offering.

Gemalto Sentinel®Cloud Services

Sentinel Cloud Services make it quick and easy for SaaS providers to define their service catalogs, provision and control user access at the feature level, measure service usage for billing and business intelligence purposes, and instantly adapt their service catalog to embrace new and evolving market opportunities.



DEFINE

A versatile service catalog for maximum market penetration

PROVISION

Contracts & automate business processes to maximize efficiency

CONTROL

Service authorization to ensure service agreement compliance

MEASURE

Service usage for billing support & instant business insight








ADAPT

Service offering on-the-fly to embrace new & evolving markets

About Gemalto's Sentinel Software Monetization Solutions

Gemalto, through its acquisition of SafeNet, is the market-leading provider of software licensing and entitlement management solutions for on-premises, embedded and cloud-based software vendors. Gemalto's Sentinel is the most trusted brand in the software industry for secure, flexible, and future-proof software monetization solutions.

Join the Conversation

-  > Facebook
www.facebook.com/licensinglive
-  > LinkedIn
bit.ly/LinkedInLicensingLive
-  > Twitter
twitter.com/LicensingLive
-  > Google+
plus.google.com/u/2/106533196287944993975/posts
-  > Sentinel Video Cloud
sentinelvideos.safenet-inc.com/
-  > Blog
<http://www.licensinglive.com/>
-  > Sentinel Customer Community
sentinelcustomer.safenet-inc.com

Contact Us: For all office locations and contact information, please visit www.safenet-inc.com

Follow Us: licensinglive.com

 GEMALTO.COM

gemalto
security to be free