White Paper on Building a Scalable and Profitable SaaS Business Model

techcello

Contents

Introduction

With the increasing trend in on-demand consumption, SaaS business model has started becoming the de-facto standard for delivering software products. While this is a great boon for ISVs to tap the SMB segment, SaaS business model also brings inherent challenges to the table. On a broader scale, all these challenges can be summed in two words – "Scalability" and "Profitability". ISVs that have understood and acted on these two challenges are the ones that are able to survive on the long run. In this white paper we will be discussing the challenges and solutions involved in building a "Scalable" and "Profitable" SaaS Business Model.

Challenges in SaaS Business Model

Let's start with some insights on some of the unique attributes of a SaaS business model that poses challenge for ISVs.

Scale

One major difference between a traditional perpetual model and SaaS is the fact around the number of customers that are required. Let's take a look at an example to understand this in greater detail.

Perpetual Model - Yearly Target of 600K USD

Month	New Customers to be Won	Cumulative Total Customers	Avg. Deal Size	Monthly Revenue	Cumulative Total Revenue
1	1	1	$ 50,000	$ 50,000	$ 50,000
2	1	2	$ 50,000	$ 50,000	$ 100,000
3	1	3	$ 50,000	$ 50,000	$ 150,000
4	1	4	$ 50,000	$ 50,000	$ 200,000
5	1	5	$ 50,000	$ 50,000	$ 250,000
6	1	6	$ 50,000	$ 50,000	$ 300,000
7	1	7	$ 50,000	$ 50,000	$ 350,000
8	1	8	$ 50,000	$ 50,000	$ 400,000
9	1	9	$ 50,000	$ 50,000	$ 450,000
10	1	10	$ 50,000	$ 50,000	$ 500,000
11	1	11	$ 50,000	$ 50,000	$ 550,000
12	1	12	$ 50,000	$ 50,000	$ 600,000

Let' say an ISV is operating in one-time licensing model and the average deal size is 50,000 USD. Assuming the license target for the year is 600,000 USD, you will need 12 customers to meet this target, which transforms to 1 customer per month. Please refer the above table for the workings.

Let's assume the same product is now offered in a subscription model with an average monthly subscription of 2,000 USD per customer per month. In this case, you will need 25 customers from day 1 to meet the target. Not just that, but you will also have to deal with customer churn (which is not applicable for perpetual model) by compensating (adding more new customers) the number of customers you lose every month. Therefore, you need to add 6 new customers every month to fill the gap created by last month's

churn (considering 25% annual churn). As you could see from the "Cumulative Total Customers" column from the below table, you would need a total of 36 customers during the course of the year to meet the same target.

SaaS Model - Yearly Target of 600K USD

Month	New Customers to be Won	Cumulative Total Paying Customers	Customer Churn	Cumulative Total Customers	Avg. Deal Size	Monthly Revenue	Cumulative Total Revenue
1	25	25	1	25	$ 2,000	$ 50,000	$ 50,000
2	1	25	1	26	$ 2,000	$ 50,000	$ 100,000
3	1	25	1	27	$ 2,000	$ 50,000	$ 150,000
4	1	25	1	28	$ 2,000	$ 50,000	$ 200,000
5	1	25	1	29	$ 2,000	$ 50,000	$ 250,000
6	1	25	1	30	$ 2,000	$ 50,000	$ 300,000
7	1	25	1	31	$ 2,000	$ 50,000	$ 350,000
8	1	25	1	32	$ 2,000	$ 50,000	$ 400,000
9	1	25	1	33	$ 2,000	$ 50,000	$ 450,000
10	1	25	1	34	$ 2,000	$ 50,000	$ 500,000
11	1	25	1	35	$ 2,000	$ 50,000	$ 550,000
12	1	25	1	36	$ 2,000	$ 50,000	$ 600,000

This is almost 3X times increase in the customer base compared to the perpetual model. The only way to keep the gap created by churn to a minimum is to retain existing customers.

Therefore, the challenge is to acquire more customers in a short period of time. This is one of the reasons why tapping SMB (Small and Medium Business) segment of customers becomes extremely critical for ISVs. The other challenge is to ensure the customers are retained and churn is minimal.

Revenue

From the earlier example, you will be getting only 2,000 USD per month from the same customer that was earlier providing you a lump sum of 50,000 USD. This scenario completely changes (rather challenges) your spending power. In addition to this, since your primary target will be SMB segment, you will have to provide attractive and flexible pricing models to create interest.

Infrastructure

Traditional Perpetual licensing models had the benefits of "shelfware" , where the ISV need not worry about the infrastructure aspects of how the product is installed and managed. But with SaaS it's the other way round where the customer need not worry about software or hardware aspects, as its falls under the responsibility of ISV. This is a big and new responsibility on the shoulders of an ISV. Not to forget that this infrastructure need is only going to keep increasing as you get more customers. In addition to this, the security and availability of the product will be the standard questions posed by potential leads, and having a convincing answer is required to move them forward in the sales cycle.

Operations

Service aspects of a SaaS business model is defined and controlled by the SLA terms. The entire organization has to be aligned to meet these SLAs. With the multi-fold increase in customer base, the ticket volumes will also increase proportionately, which means more work for your support team. When it comes to collecting payments, what used to be a simple job of 1 invoice for 50,000 USD now becomes 2,000 USD every month for every customer. Once again more work.

Considering these unique challenges, the question to be asked is "How **complete** is your SaaS Solution in meeting all the challenges/needs that are arising due to the unique nature of SaaS model?"

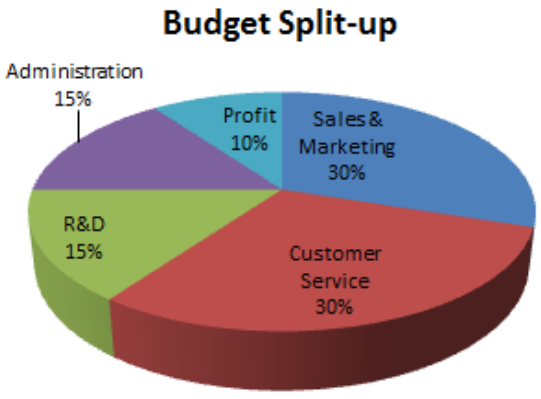
Understanding SaaS Life Cycle

Blind Spots in SaaS

When ISVs decide to build or migrate their existing SaaS solution, below is the typical phase they go through, which starts with an idea or an existing product, followed by a technical feasibility or a PoC, followed by architecture, development, testing and deployment (on cloud). Once it's deployed ISVs think the "Solution is ready" and they are on "SaaS".

But in reality, ISVs typically tend to overlook two other important areas – "Selling" and "Servicing". But you may ask "What has my solution got to do with selling and servicing?" The answer is quite evident from the fact that these 2 factors straightaway determine your profitability. How? Selling brings money to your pocket. Servicing takes it out. So, the net is left on the table. Therefore, when we talk about a "complete" solution, it should have certain aspects in it that can help in the "selling" and "servicing" areas of your SaaS business.

SaaS – Number Game



Monthly	Avg. Revenue per Customer	Number of Customers *	Total Revenue	Cost of Delivery (CoD) (30%)
1	$ 2,000	10	$ 20,000	$ 6,000
2	$ 2,000	11	$ 22,000	$ 6,600
3	$ 2,000	12.1	$ 24,200	$ 7,260
4	$ 2,000	13.3	$ 26,600	$ 7,980
5	$ 2,000	14.6	$ 29,200	$ 8,760
6	$ 2,000	16.1	$ 32,200	$ 9,660
7	$ 2,000	17.7	$ 35,400	$ 10,620
8	$ 2,000	19.5	$ 39,000	$ 11,700
9	$ 2,000	21.5	$ 43,000	$ 12,900
10	$ 2,000	23.7	$ 47,400	$ 14,220
11	$ 2,000	26.1	$ 52,200	$ 15,660
12	$ 2,000	28.7	$ 57,400	$ 17,220

*10% Month on Month Growth

Let's look at a typical budget split-up for an ISV in SaaS model. Sales & Marketing constitutes 30%, Customer Service constitutes 30%, R&D constitutes 15%, and Administration constitutes 15%. That leaves a 10% profit margin. Customer service includes all the cost for delivering the service to customer, which typically includes hardware, license cost, tools cost, tech support and account management. It's interesting to note that even large players like Salesforce.com operate at 30% Cost of Delivery (COD), which means for smaller players this number could be even lower.

Let's apply these % on a sample data to get a feel of what budget numbers can be expected. The above table shows the revenue figures by month. Considering average revenue per customer as 2,000 USD and 10% month on month growth, the budget availability for delivering your complete SaaS services ranges from 6,000 USD to 17,000 USD. As you can imagine, these are pretty tight numbers to manage.

Therefore, it's clear that SaaS companies are going to be running a tight budget and still have to dazzle their customers with superior service, at the same time keeping the costs low.

SaaS Life Cycle

This picture gives you a holistic view of the various stages or phases that happen in a typical SaaS business environment. It starts with creating your product subscriptions, against which your customers may register or subscribe to. Once they sign-up you need to bring them on board in to the system, which will require a series of configurations, setting up of security measures, getting users on board and define their access rights. Once this done, you will constantly monitor them for usage, audits and specific actions like data backup. For each subscription period, you will have to raise the bill and collect the payments from the customer. Finally, you would like your customers to upgrade or renew their subscriptions.

This being the life cycle of a SaaS model, let's look what the needs are from various stakeholders to execute this life cycle.

SaaS Life Cycle management Solutions – Needs

We will look at three important stakeholders within the ISV.

Management Team – what they need is an error free and efficient mechanism through which they can provide innovative subscription models, pricing models, etc.

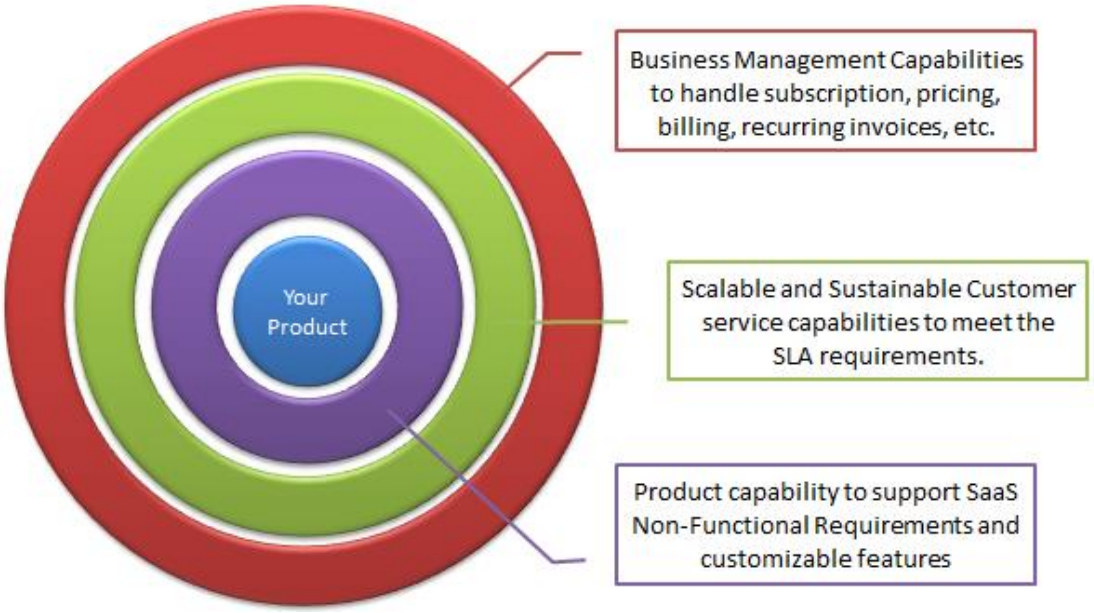
Customer Service Team – They need a quick and cost effective way to roll-out new customers. They will also need a way to tackle increasing number of product support tickets and monitor the product so that they can take proactive steps.

Product Engineering Team – Their single focused goal is to build a high speed system, which is highly scalable and secure.

Do these needs differ from ISV to ISV? Not really. The SaaS life cycle and the needs do not talk anything about the product itself, rather it talks about how the product is managed and delivered. Therefore, these needs pretty much remain the same between ISVs.

Cello – A Quick Preview

Cello – One Stop Solution



Around three years back, Cello identified the needs of an ISV in order to service the end to end SaaS life cycle. For any product that is planned for a SaaS model, Cello provides these 3 additional rings around it, which takes care of all the SaaS Life Cycle needs that we discussed earlier.

The inner most ring, is about adding capability to your product to support Non-Functional Requirements(NFR) like Scalability, Configurability, Security, etc. and also design your product features in such a way that they are easily customizable.

The middle ring contains features that can help your tech support team in customer implementations as well as addressing their service requests.

The outer ring provides the business management capabilities to handle subscription, pricing, invoicing, payments etc.

Cello has modularized and packaged all these common needs as an out of box unified solution, which can be easily plugged in to your product.

Cello – Solution Areas

Let's see what each of the stakeholders will get from the cello integrated solution,

Management Team – They get equipped with executive dashboards and monetization tools through which they can completely control and operate the subscription and revenue collection of their product.

Customer Service Team – Most of the typical customer services requests are now automated, and hence, the service team can now process lot more requests in a much faster way. In addition to this, self-service portals also allow the end customers to address many of their simple requests without even having to come back to the tech support team.

Product Engineering Team – They get equipped with an engineering stack that supports all the NFRs required for a SaaS product. The engineering features also help in cutting down a significant amount of development time, resulting in faster time to market.

Let's delve in to the challenges in each of these three areas – Business Management, Customer Service and Product Engineering – and how cello can help in resolving them.

Business Management

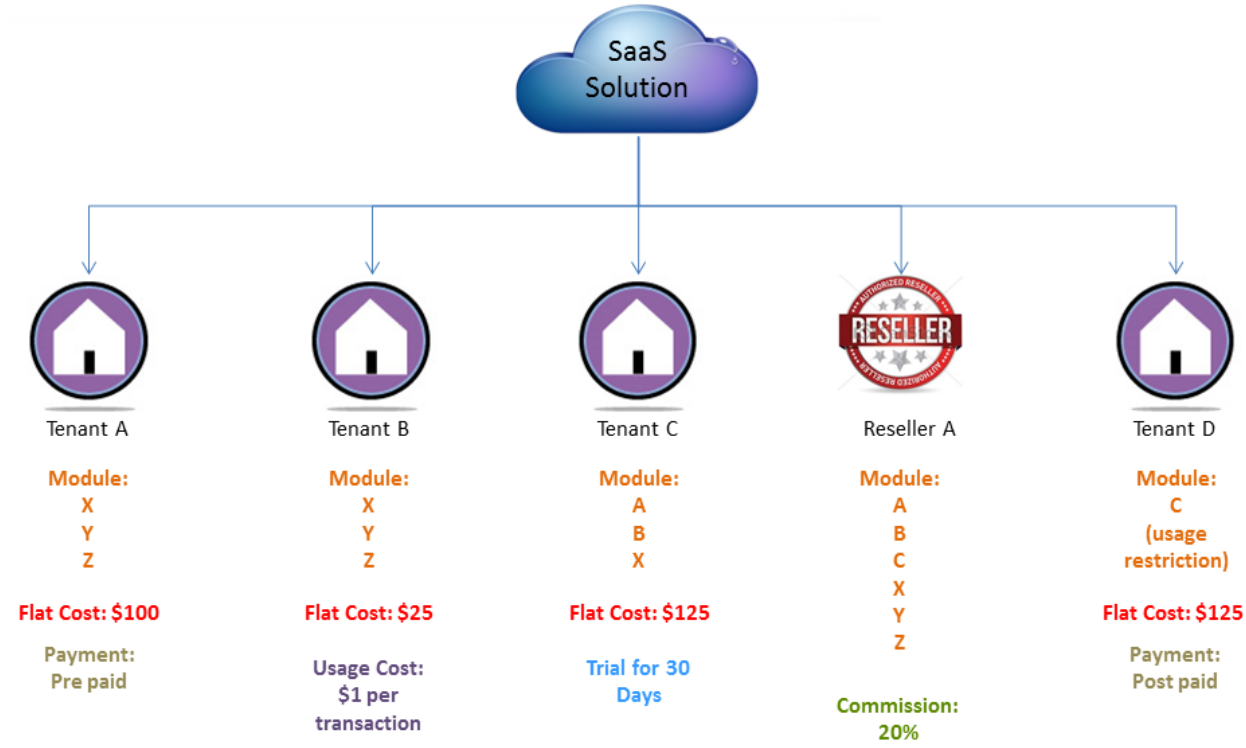
Challenges

These are the questions that any SaaS management team will be interested to find answers. As you can notice all these questions boil down to "selling" and increasing revenue inflow.

Solutions

Following are some of the industry best practices to address the above business challenges.

SaaS solutions today have reached the levels where you can sign-up and start using the trial in less than 2 minutes. Offering self-registration capability and availability of the product in live trial will help in cutting down the sales cycle. Having module wise access of the product, will allow the customer to pick and choose the modules that are required. These modules can be priced individually, which will benefit both ISV and customers. Let's look at few scenarios that can give you an idea on the variations you can bring in your SaaS Subscription model.



Tenant A – has subscribed to modules ‘X’, ‘Y’ and ‘Z’ at a monthly flat cost of 100 USD, and will be paying it in a pre-paid model (i.e. the payment will be made at the start of the month).

Tenant B – has subscribed to the same set of modules just like Tenant A, but will be paying on a usage based model. It will be a monthly (base) flat cost of 25 USD plus 1 USD per transaction, which will be done through the SaaS system. For example, if this tenant has done 53 transactions in a given month, then they will be paying 25 + 53 = 78 USD on a post-paid model.

Tenant C - has subscribed to modules ‘A’, ‘B’ and ‘X’ at a monthly flat cost of 125 USD, but will be on trial mode for the first 30 days. So, the first 30 days of usage by this customer should not be considered for invoicing (or rather for applying cost). Your SaaS solution should be able to differentiate between a paying customer, non-paying customer, paying customer currently on trial mode and non-paying customers on trial mode.

Reseller A – this reseller has access to all modules. However, as an ISV you may want to control what modules can be available for which resellers. This is required when your solution serves multiple domains, and hence you may have domain based resellers. For resellers, you will be billing 20% of the total revenue accrued by that reseller in that month. In order to do that you should know the exact data on the revenues that have come through the tenants boarded by this reseller.

Tenant D – has subscribed to only one module ‘C’ at a monthly flat cost of 125 USD, but will be paying it in a post-paid model (i.e. the payment will be made at the end of the month). In addition to this, there is a usage limitation that has to be applied for this tenant. For example, this tenant can only perform 100 transactions per month, beyond that the system should restrict the transactions.

The above scenarios are just few examples to show the variations (or rather innovations) you can bring in to attract more customers. One of the key innovation area would be the value based pricing models, as shown in case of Tenant B, where the billing was done based on the value the customer could yield from your solution.

It’s very important to increase your conversion rate i.e. from trial customers to paying customers. In order to do this, you need 2 important data points. #1 Hot Trials (or leads) recording a high usage of your system, which means they are in the happy path #2 Trials recording low usage of your system, which means they are either not happy about what they see or is stuck up with some road block.

Keeping track of who is using and who is not using your product will help you in taking proactive measures. Your sales team can go all out on the hot trials – those using the system. At the same time, you can also enquire with those trials that aren’t going well, to find out if they are experiencing any difficulty and offer help.

As you can see from the above scenarios, there are several attributes (color coded in the above picture) that have to be considered while managing the subscriptions of your customer. It’s extremely difficult, time consuming and error prone to manage these subscriptions in a manual way. As you grow, this effort will only

increase multi-fold. Therefore, having an automated way of managing the entire subscription and billing process can save significant time, effort and customer dissatisfaction (as result of billing errors).

Your invoice generation can be completely automated. Invoicing engine can find out the customer's chosen subscription plan, apply the pricing model, find the usage, and calculate the total bill amount. This can save a significant amount of time and is totally error free.

Allowing your customers to register their credit card or pay pal account for automatic debit, can also save a good amount of time for both you and as well as the customer.

Self-service customer portal provides complete transparency of the billing details to customers. Similarly, reseller portal can help you keep track of how your resellers are performing.

Cello - Business Features

As mentioned earlier, cello's business management edition encompasses all the solutions that were discussed in the earlier section. Subscription related features will allow you to publish multiple subscription plans, with multiple price plans. You can also have any combination of pricing model (fixed or usage cost), which can vary from tenant to tenant.

Customers can self-register and start using the product within no time. Customer management features provides you with customer analytic information to handle your leads and customers. Invoice generation is completely automated with support for metering and proration. For payments, all major credit cards and paypal account is supported.

Customer Service

Customer service is one of the very important functions of a SaaS business. Customer service team has a bigger role to play in reducing customer churn rate. As per industry research, in any business around 65% of the customer loss is due to the improper customer service. Especially in a SaaS model this customer loss called "churn" could prove devastating.

While having a world class customer service team is important, it is also important to note that in order for the SaaS business to be profitable, it is extremely necessary to curb the cost lesser than 30%.

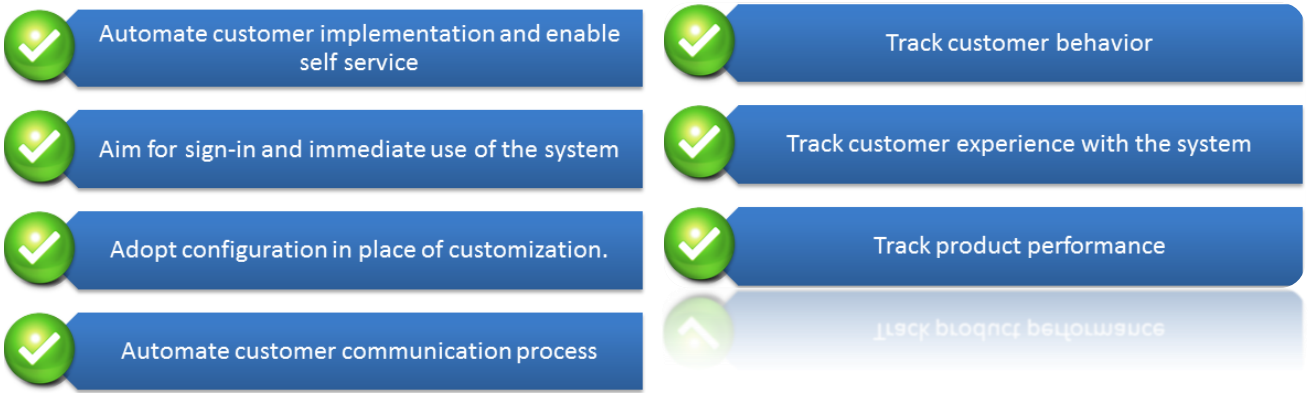
This 30% should also cover the expenses of hardware, software licenses and so on which then boils down to lesser than 20% available to be spent for customer service. Considering the fact that SaaS is going to be a high volume game, this leaves us with the customer service team handling many customers and if not optimized, can lead to highly increased cost and timeline.

Challenges

The customer service team whose primary responsibility is to onboard new customers and service the existing ones constantly try hard to find solutions for the following challenges.

Solutions

Following are the some of the best practices that can be adopted to address the above challenges.



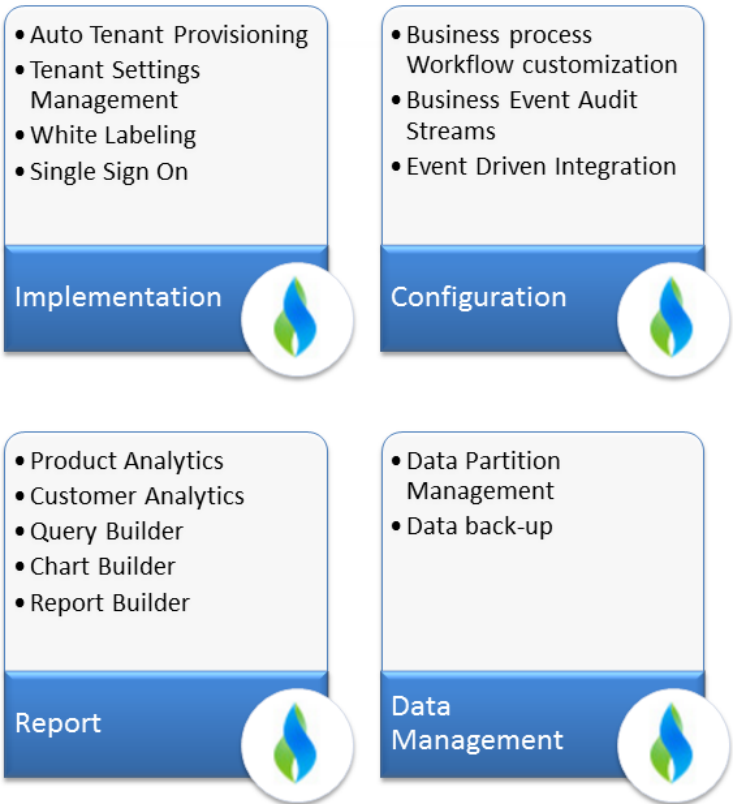
Enable your SaaS solution with self-servicing (including self-registration) capabilities so that many of the simple requests like password reset, user lock/unlock, permission changes, workflow changes, etc. can be done by the end user (tenant admin) themselves. Customization in SaaS is the last thing you may want to do. Managing multiple customized solutions is going to be difficult and expensive. Instead of customization, you need to embed configurability all through your product so that it can be effectively used during implementations. There are several email communications that have to happen with the customer, for

example: registration email, activation email, trial expiry reminder email, credit card failure email, etc. It's important that these communications are automated and available as part of the system.

Assume a scenario, where one of your customer calls up and expresses their concern that your system is occasionally slow. There could be several reasons why the customer may experience this issue, including those that are not in the control of the SaaS service provider. However, you will have to be sure before you can even respond back to the customer. In a typical scenario, this request gets forwarded to the engineering team and they try to go through the logs and other details, and at times they may not even be able to validate this information. Net result – loss of engineering team's time, tech support has to spend more time explaining to the customer, and finally the customer is not happy.

The above scenario can be totally avoided if you are able to track your customer experience and product performance. Product analytics can throw you information like average response time experienced by a given customer over a period of time. Now your tech support person can quickly pull up this chart and find out if the response times were poor on any particular day. Not just that, but they can also drill down and find out that one particular page is taking quite a lot of time. This precise information goes to the engineering team, and they can quickly find out the root cause and fix it. Now, in this scenario the customer is going to be extremely delighted with the quick response that you can provide, with minimal effort from your end.

Cello – Customer Service Features



Cello's customer service related features exactly provides the above solutions right out of the box. It offers the following modules,

Completely configurable multi-tenant workflows which allow the tenants themselves or the customer administrator to alter the workflows to the way they need via user interface. Configurable forms which allows the tenants themselves or the customer administrator to go and create adhoc forms or extend the existing forms. White Labelling which allows the tenants to configure their themes and logos. Configurable Reports which allows the tenants and the customer administrator to quickly build reports from any application data.

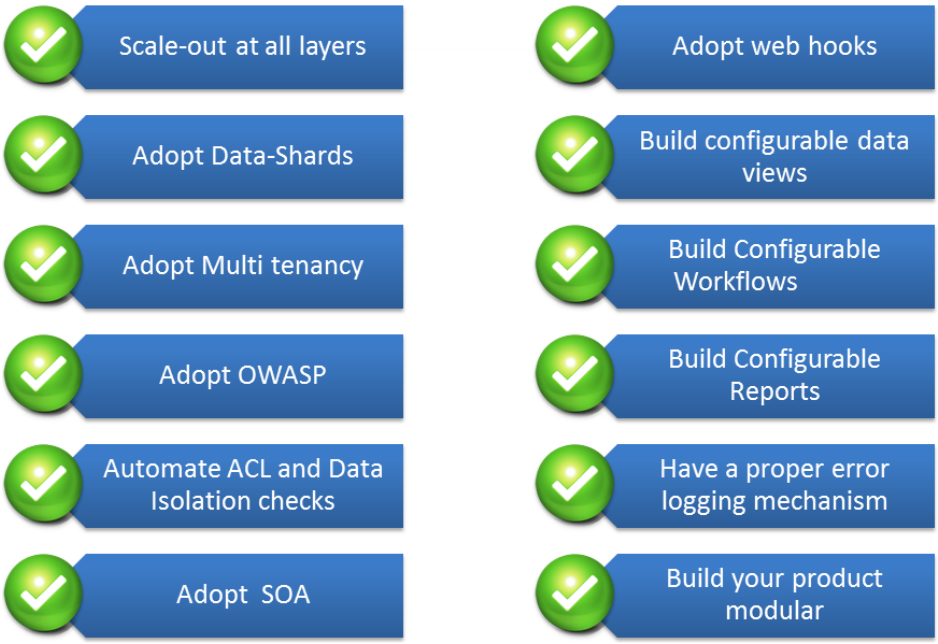
Product Engineering

Challenges

Following are the top concerns of most of the SaaS engineering team. Most of these factors are not a big concern in on premise environments as it caters to a single customer and hence, load is not going to be extremely high. Moreover it is delivered from within the same network and hence, the threat of attack is not high and integration can be simple as the (connecting) solutions are also going to be available together within the same network.

Solutions

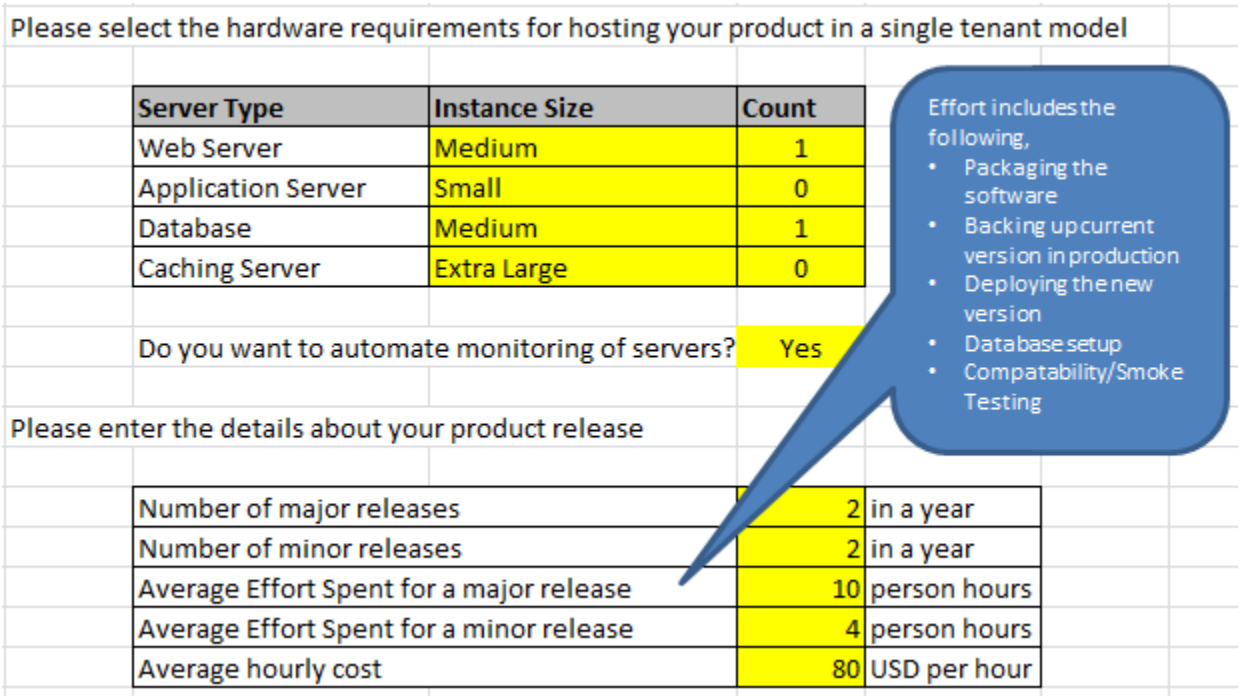
SaaS model not only demands the business functionality to be addressed correctly but also deliver in such a way that it can be highly scalable, secure, extendable and also easy to configure and integrate. Following are some of the best practices that should be followed in engineering your SaaS product.



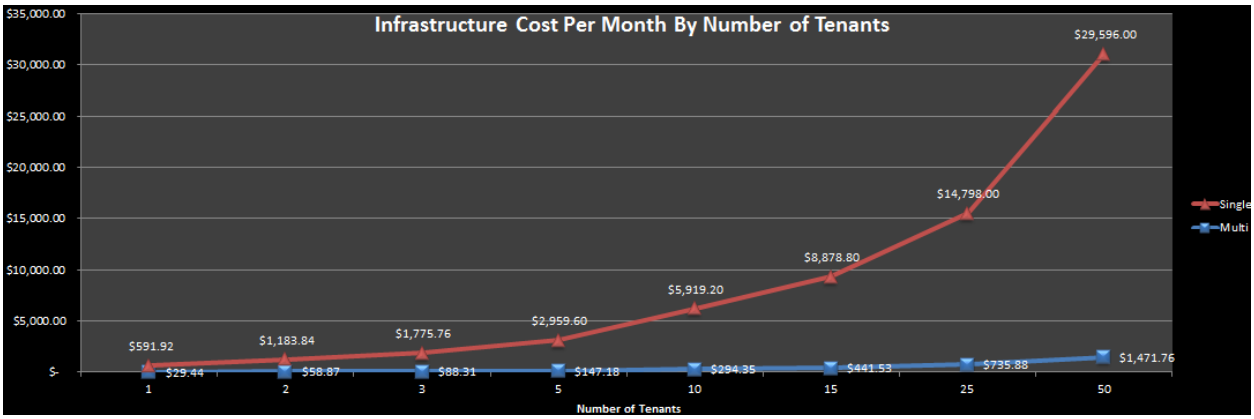
In today's cloud infrastructure scenario, scaling at the web and application layers is extremely easy particularly with technologies like java, .NET, which inherently supports scaling. However, scaling the database layer is the most difficult task, which can be simplified by adopting Data-Sharding as a principle right from the architecture. Data sharding is basically partitioning your data in some logical groups, where each group can be individually scaled out. For example, you can have one group (sharding) to hold all US customer's data, and one group to hold all UK customer's data and one group to hold the remaining customer's data. Sharding also allows you to totally move these groups to different physical (database) servers so that they can be processed more efficiently without any bottlenecks. The good thing is, even when your US customers grow heavily you can further split the group as US-East, US-Central and US-West. This way you can achieve unlimited scalability with zero engineering effort.

Multi-tenancy can have a great impact on the scalability and the profitability aspects of your SaaS business. Let's understand what multi-tenancy can bring in through a simple tool – Single vs. Multi-Tenant Calculator.

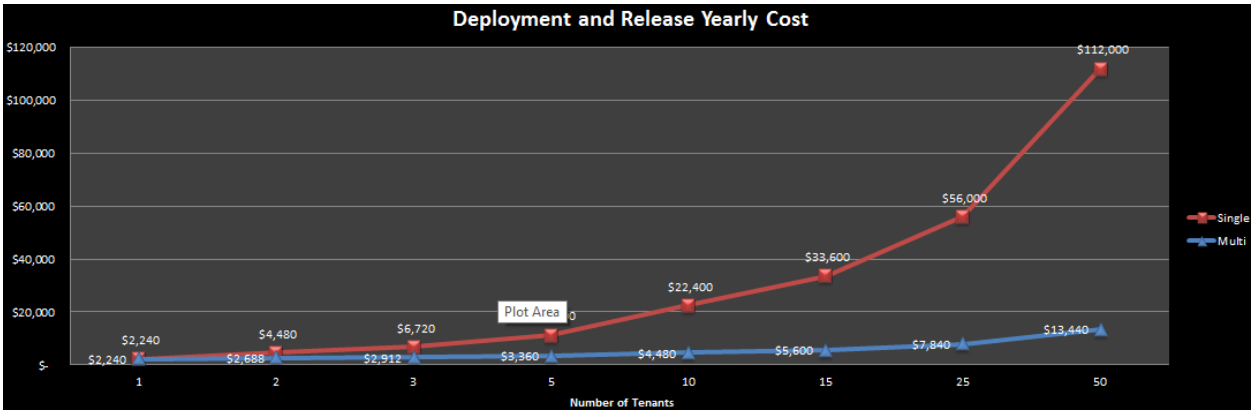
Single tenant model typically requires the entire product to be setup in an isolated environment and accessed by the tenant. In this case you will have to dedicate the servers for that particular tenant. The tool collects the hardware requirements for hosting a product in single tenant model (i.e. for one customer). It also allows you to enter the details about your product release.



Once you enter the details the following reports are generated. The below report shows that the infrastructure cost per month for 1 tenant in single tenant model will be 591 USD, whereas the same in multi-tenant model would be 29 USD. Similarly, when you reach 50 tenants, single tenant model would cost you 29,596 USD per month, compared to 1,471 USD per month in multi-tenant model. As you can see this is a significant cost saving for an ISV, which can directly reflect in your profitability.



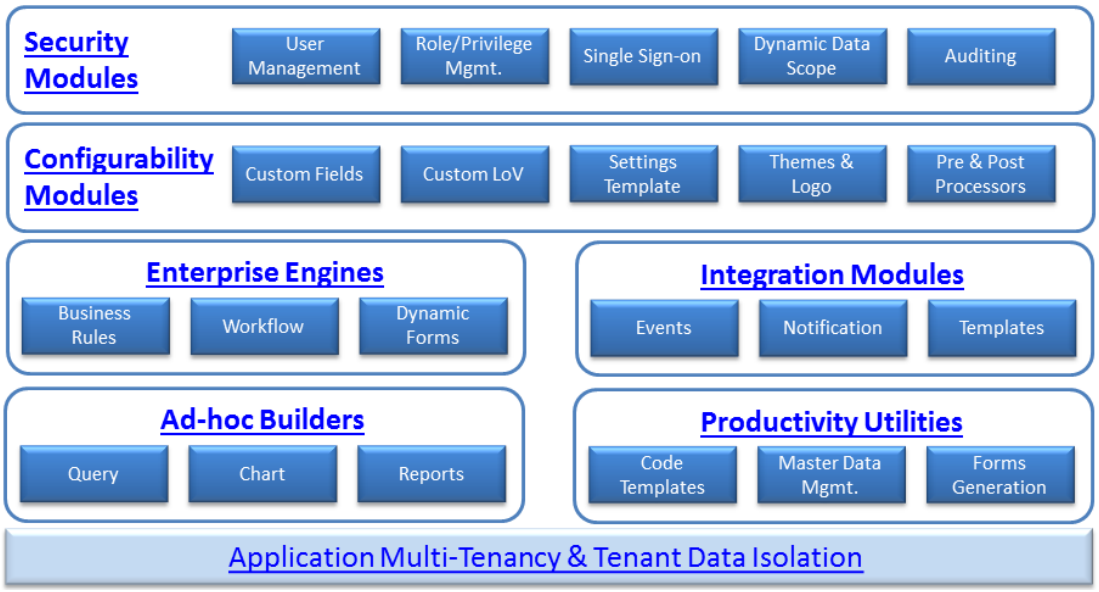
Similarly, the below report will throw light on the widening gap in terms of cost difference between single and multi-tenant way of managing deployment and release. As you can see, when you are running at 50 tenants per year, you would have spent 112,000 USD compared to 13,440 USD for managing the deployment and release activities.



Security is one of the primary concerns of customers while moving to a SaaS model. Any issues with data crossing the tenant boundaries can lead to business disaster and hence, it's very important to build strong line of checks for data isolation and access control right at the framework level.

Cello – Product Engineering Features

Following are the features supported by Cello to address the product engineering needs.



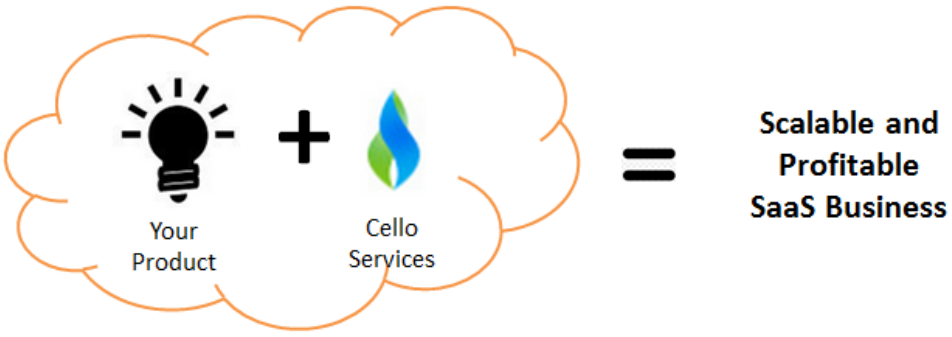
Cello – One Stop Solution for All Your SaaS Needs

Addressing the Business, Customer Service and Product Engineering areas not only brings in loads of benefits to an ISV but more importantly it can allow you to surpass your competitor by leaps and bounds. However, building all these features not only consumes more time and effort, but also requires great expertise. While it's important to bring these capabilities early in to your SaaS solution, it may significantly delay the start of (business functionality) development. This puts the ISV in a tough situation of choosing between a long term strategic goal of having a perfect system vs. the short term goal of hitting the market earlier.

Cello can help in achieving both the short term and long term goals of ISVs. More importantly Cello is available as one integrated and unified solution that can be easily plugged in to any existing or new SaaS product. Alternatively, you can also use Cello as a base framework to build your product, in which case you can save almost 50% of your engineering budget and time.

Following are some of the key benefits from Cello,

- 10X times reduction in operating expenses
- 40% increase in trial conversions
- 50% reduction in implementation time and cost
- Increase in customer life time value
- 50% savings in your engineering budget
- 60-80% reduction in support workload
- Reduce customer churn
- On time Payment collection
- Accurate and Transparent Billing
- Increased product value



Summary

Building a successful product requires a lot of focus on the domain, solution, usability and innovation. Cello clearly understands this and enables ISVs to focus on the product by taking care of all the needs of a SaaS business model.

We hope this white paper helped in explaining our perspective of Building a Scalable and Profitable SaaS Business. Please feel free to email your feedback at info@techcello.com

Please visit www.techcello.com and www.slcm.techcello.com to know more about Cello.