CUSTOMIZATION VS. CONFIGURATION

Best Practices for Choosing a SaaS Application

AppointmentPlus™

AppointmentPlus.com

Table of Contents

Introduction

Being an enterprise IT professional is a lot like being a tightrope walker.
Let me explain with a story:

> *Bryan is the IT director at a national retail automotive services company. He's in the break room to grab his morning cup of coffee and . . . yes! Someone brought donuts.*
>
> *He's deciding between a cinnamon twist and a jelly when in walks Janet, the company's VP of operations.*
>
> *"Hey Bryan," she says. "I found a tool that I want to implement for the business."*
>
> *Bryan turns away from the donuts. (He didn't need the calories anyway.)*
>
> *Janet goes on. "I started working with a personal trainer, and he has an online scheduling application that I can use to book appointments with him. It's great for me, because I get reminders of upcoming appointments, and the option to add on other services at the gym. And now that my trainer isn't scheduling his appointments manually, he says it's saving him five hours a week. Plus, he gets stood up less often, so his income's increased."*

Bryan knows what's coming. He tries to look excited.

"So," Janet continues, "can you have a short list of recommended online appointment scheduling solutions on my desk by next week? I love the idea of being able to trim costs by implementing a service that will improve the customer experience. I went ahead and made a list of all the features that are must-haves for me."

She hands Bryan a list that is a full page long, 10-point type, single-spaced. "I'm sure John in Product and Sarah in Customer Support will want to weigh in with their requirements too."

If you identify with this scenario, you're not alone. According to a 2014 Gartner study, spending on cloud services totaled around $80 billion in 2014, and is projected to reach $153 billion by 2018. The study also concluded that while CIOs tend to focus on SaaS benefits like operational agility and competitive advantage, the rest of the business is primarily interested in SaaS for the cost savings.[1]

Online appointment scheduling software is an example of a SaaS solution that delivers all of these benefits. So it's worth looking into. But when all the Janets and Johns and Sarahs get together, the list of "must-have" features grows like a five-year-old's holiday wish list.

And so does Bryan's aspirin supply order. Because the more custom features he has to find in one application, the greater the headache. And the greater the expense.

So it falls to IT to help the business leaders understand which features can be configured within the application, which need to be customized, and which can be implemented later (or not at all). It's up to the Bryans of the world to balance the demands of the business and the realities of IT.

To step out onto the tightrope, and make it to the other side.

It sucks to be Bryan

If you're the Bryan of your company, then you're right with me. If you're not — if you're Janet, or her company's CMO or CFO — you're more apt to be thinking, "Tightrope, yeah. And why am I interested in this analogy?" Fair question.

The answer is that every step of Bryan's slow, considered, agonizing walk along that tightrope takes him farther off task. He's no longer doing the work you're paying him to do. Instead, he's off conducting research and interviewing SaaS companies, trying to find "the one" that offers all the features on that gigantic wish-list.

His team is getting mired in his backlog of projects, which starts to impact other departments. And you can see how one IT manager researching an enterprise-wide improvement like this can start to wreak havoc throughout the company.

It's all totally *unnecessary havoc*, by the way. Hence this eBook.

Let's talk about SaaS, baby

Forget about that long list of must-haves for a moment. What you're looking for at its most basic level is a SaaS solution. And the true beauty of SaaS is that it can be modified to meet your specific business needs, without — and this is key — branching from the code base.

What you'll realize by the time you've finished reading this paper is that there is a shortcut. A way to ensure all of Janet's and John's and Sarah's scheduling software dreams come true, without having to compare long lists of features and benefits, host multiple vendor presentations, or drive your team to mutiny.

Think of this eBook, then, like one of those hidden harnesses that help stabilize the tightrope walker.

What you'll learn;

1. The three ways in which a "pure SaaS" online appointment scheduling application can be modified to meet the needs of the business
2. Why none of those three ways involves branching the code, keeping in line with the SaaS best practice of maintaining a single code base
3. How branching the code negates many of the benefits of SaaS
4. The truth that not all online appointment scheduling SaaS vendors follow the single-code-base best practice
5. How you can toss that list of must-haves, and still make everyone happy.

3 ways to make a SaaS application “your own”

Many of the benefits of Software-as-a-Service come from the fact that SaaS is built not specifically for one client, but generally for a whole bunch. That said, most enterprises can’t just take the application “off the shelf,” and need to make it their own in one of the following ways:

1 Configuration

The best SaaS applications are highly configurable. They’re built with a robust set of configuration options that affect the functionality as well as the look and feel of the application. You can make the application your own by manipulating existing settings or preferences. No IT needed (no offense, Bryan). No code changes. Anyone who can navigate the admin panel can configure the application. Some might say configuration is so easy a monkey could do it. (Of course, we would never say that!)

Now, writing custom code isn’t bad, as long as it’s done within the single code base. But it does involve professional services fees and time. So ideally, you find a SaaS vendor with a highly configurable application, so that most of your needs can be met by manipulating existing settings or preferences. Then you reserve the custom coding for those few very unique features that really are must-haves.

2 Customization

The second way to make the application your own is through customization. There’s a long- running debate about configuration versus customization in SaaS. A lot of people feel very strongly that customization has no place in a SaaS environment, because it necessitates branching from the primary code base. And branching completely wipes out many of the benefits of SaaS.

But here’s where they’re wrong: The customization itself isn’t what negates those SaaS benefits. It’s the branch that creates the problem.

**Customization is not a four-letter word —
if it’s done within the code base.**

For those few unique features that really are must-haves, but which aren't available through simple configuration, a pure SaaS vendor will **write custom code within the code base** to enable that functionality. Adding the customization within the code base allows the vendor to deliver the functionality the business needs while still maintaining that single code base — and retaining all of the benefits of pure SaaS.

It's a SaaS best practice.

As Salesforce explains it, "Because of the way SaaS is architected, these customizations are unique to each company or user and are always preserved through upgrades. That means SaaS providers can make upgrades more often, with less customer risk and much lower adoption cost."[2]

3 API

The third way in which a SaaS application can be made your own is through the vendor's API. When you're looking for very in-depth or complex custom functionality, customizing the base code might not be the answer. In this case, you'll want to consider building a custom application on top of the API. Or, developing a custom-built (non-SaaS) software solution.

A custom application built on top of the SaaS vendor's API is neither a customization of the base code nor a branch from the base code. And compared to a custom-built software solution, it has advantages.

For one, the application interacts via the API with the SaaS application's code base, so you still benefit from the application's functionality and its upgrades.

But, on the downside, you are responsible for maintaining the custom application, and QA testing against new releases can be a challenge. (Though that is also the case with custom-built software solutions.)

Making the SaaS application your own by building a custom application on top of the API creates a sort of hybrid between pure SaaS and custom software. On the pure SaaS side, you retain the benefits of quick bug fixes and feature upgrades that the vendor rolls out through the code base. On the custom software side, you get those unique features the business has determined are indeed “must-haves.”

How our clients make AppointmentPlus their own

There are three ways that our clients modify the AppointmentPlus application to meet both line-of-business and IT needs. None involves branching from the code base.

1. Configuration

The AppointmentPlus application is highly configurable and intelligent. Rules-based routing and skills-based scheduling ensure it fits each client’s business processes. Our clients modify the application to fit their needs, and they do it easily by manipulating existing settings and preferences. No code changes are involved. **See our available functionality: Schedule a call with an Enterprise Partner Manager.**

2. API

Through the AppointmentPlus Scheduling Cloud™ API, our subscribers can pull appointment, customer, staff, and services data into other applications. In cases where a client needs functionality beyond what’s built into our application, or through integration with another application, we build (or work with others to build) custom applications on top of the API. See for yourself: **Download our Developer’s Guide.**

AppointmentPlus
Mon, October
Sharon Davis - General...
Penny Pruit - Specialized...
Lunch
Rick Rogers - Extended...
Today
Day
List
Month

3. Customization within the code base

In the cases where neither existing system parameters nor API integration meet the client’s needs, we provide customization within the code base. Such development is undertaken according to an agreed-upon professional services contract and detailed scope of work.
Flip to page 15 for a real-life example of pure SaaS in action — one delivering a 2640% ROI.

Getting the C-suite to narrow down their wish list

No matter which one of these three routes you choose, none is going to be capable of delivering everything on your business leaders' long, long wish lists.

Experience may tell you the only way to get everything you want is to build a custom application in-house. But your experience will also remind you the business isn't going to high-five you for getting them into a multi-year project with costs well into the millions.

Solution: Toss it to your vendor!

A great SaaS vendor will join you on the tightrope. They'll be right there onsite or via conference call to help you explain to the business the tradeoffs between custom features and functionality on one side, and time, money, and complexity on the other. When the party who stands to gain the most advises against incorporating some of those more complex and costly features, executives usually listen.

A little experience with the product can also help here. After getting the application into production, a capable and qualified SaaS vendor will assess how people are using it, and report back to you. At that point, it often comes out that some of the features and functionalities that were theoretically critical actually aren't all that important.

Bottom line, you can't really know what matters until people start using the application.

SaaS customization ≠ branching from the code

Maintaining a single code base is a best practice among pure SaaS providers. It's no easyfeat, but it's arguably the most important characteristic of a pure SaaS application.

> "Leading Web applications run on a single code base and infrastructure shared by all users. . . . [Otherwise,] the internal inefficiencies of maintaining a separate physical infrastructure and/or separate code lines for each customer make it impossible to deliver quality service or innovate quickly."[3]

Branching the code negates many of the benefits of SaaS

You know well the primary benefits of SaaS, and your line-of-business leaders probably do too:

- Lower total cost of ownership
- Faster time to market
- Continuous improvement

And there are lesser-known benefits too, including a smoother transition to mobile, and easier utilization of emerging technologies. (And, as luck would have it, they're all spelled out in another one of our nifty whitepapers, ***You're Not Ready for SaaS.*** It's free too.)

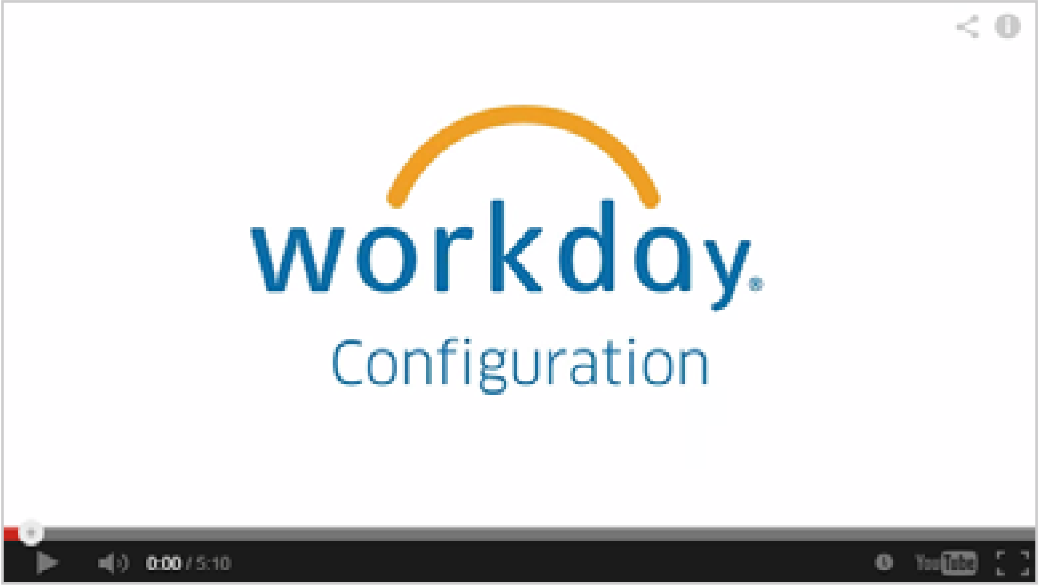
Those benefits are all great reasons to choose SaaS over on-premise software for your online appointment setting solution. But here's the rub: When the SaaS provider branches off the single code base and writes custom code just for you to enable those 1,001 unique features your line-of-business leaders "must have," the vendor is negating many of the benefits of SaaS.

When customizing the SaaS application means branching the code, then it ceases to be pure SaaS. And that means it can't deliver on the benefits of a pure SaaS platform.

Goodbye, innovation; hello, obsolescence.

Multimedia Break!

Stand up, stretch, take a bio break, or grab a cup of java, then check out this great video from HR SaaS vendor Workday, featuring technology platform advisor Naomi Bloom:



"There are some products labeled 'SaaS' which cannot be [easily] configured; they really are very restraining. And, there are some products labeled 'SaaS' that allow you to customize by breaking the code. Well, we did that with on-premise. And any of you who have been through an upgrade of a major on-premise application know that breaking the code and dealing with those customizations is absolutely not in your best interest.

"So if the question is 'Can we branch the code?' The answer is 'No.' Any vendor who says otherwise isn't a pure SaaS vendor. And without pure SaaS, you don't get the benefits."

– Naomi Bloom

Lower total cost of ownership? *Not with a custom branch*

With SaaS, upfront costs are relatively minimal. The bulk of the cost of the application is spread out over time and across users in the form of monthly subscription fees. But when a SaaS vendor creates a custom branch from the code base, it monopolizes development resources — specifically, money and time. In the end, you could end up paying as much for the custom branch as you would to implement on-premise software.

Oh, but wait. There's more: You're on the hook for maintaining the system. And when the SaaS vendor releases a new feature, you have to pay all over again for that new feature to be added to the custom branch.

Every code change is a billable activity.

And that's no small concern. *InformationWeek's* article on the customization versus configuration debate points out that, "In the case of major upgrades, customization can be just as demanding and time-consuming as migrating to a whole new application, as broken functionality may have to be redeveloped from scratch."[4]

Continuous innovation? *Guess again*

One of the chief benefits of SaaS is continuous innovation. Getting updates and upgrades is kinda the point.

> "Unlike traditional or hosted software vendors, [pure] SaaS vendors only have to maintain one version of the software and can upgrade all customers at the same time, often several times a year. . . . This faster pace of innovation ensures the application is constantly improving so customers can be confident they have the latest features at their disposal that will drive the most value for their business."[5]

But delivering frequent updates and upgrades to every client is only manageable when every client is running on the primary code base.

When the SaaS provider approaches customization by writing client-specific code that branches off the base code, then you are no longer party to the continuous improvements, upgrades, updates, bug fixes, and patches that a pure SaaS provider can easily roll out to all clients at once. The solution becomes much harder to scale. You become like an island, easily forgotten as the vendor focuses on improvements to the base code.

SaaS vendors thrive by continuously innovating their software and releasing those innovations to everyone running on the single code base. Running branched code means you become a sort of lone buoy out in the ocean, without a tether.

Concentrated firepower? *Um, not so much*

When the leading HR SaaS application vendor Workday went to a single code base, its VP of Technology Development wrote, "Developing and supporting a single version of Workday enables us to concentrate greater development firepower on that single version, avoiding the dilution of effort inherent in supporting multiple versions. The burden of supporting multiple versions is the largest disadvantage that on-premise software vendors face when competing with the cloud model."[6]

It's about opportunity cost. Any given SaaS vendor has only so many developers. If they're working on one-off custom branches from the code base, they're not working on the base system. Over time, they fall behind other SaaS providers who expend their resources serving the entire pool of clients with continuous innovation to the base code.

> *"With [pure] SaaS, all customers are using the same instance of code. Some modules may only be used by a certain sub-set of customers, but the code is never branched. This allows for streamlined development and release cycles that lower cost and speed innovation."*
>
> — **Ned Stringham**
> "5 Ways to Detect a Fake SaaS Business"
> 42ventures.com

Pure SaaS in Action — A Case Study

Challenge

The client is a Fortune 500 retail service provider with over 2,000 locations across the U.S. The IT team was tasked by the business to find a robust scheduling automation application that would integrate with their incumbent technologies, eCSDS/POS, store locator, and marketing platforms.

Solution

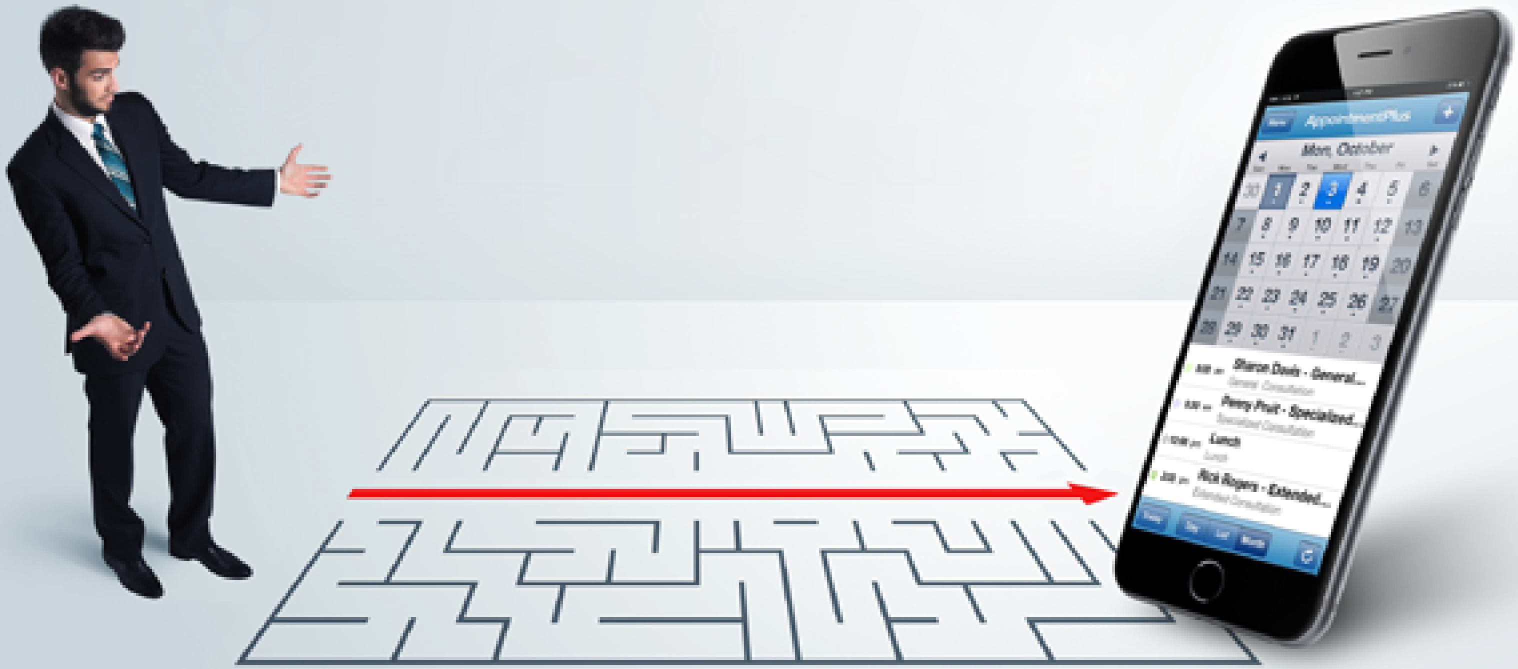
We customized the AppointmentPlus application — within the primary code base — so that it would capture our client's customer and appointment data, and then create sales orders in their eCSDS/CRM system.

Results

Keeping the application pure SaaS kept total cost of ownership down, and the client gets to take advantage of our continuous innovation. At the same time, the application actually fits the client's existing processes and systems (novel, right?). So the business can boost revenue and reduce cost. And IT can realize the full value of the systems they've already invested in.

Bottom line: The client expects to reduce the administrative costs associated with appointment scheduling by 50%, reduce the cost of no-shows by at least 30%, and increase revenue through higher customer engagement. For an all-in ROI of 2640% (and no, we didn't leave out the decimal point).

What can we do for you?


AppointmentPlus
Mon, October
Sharon Davis - General...
Penny Pruit - Specialized...
Lunch
Rick Rogers - Extended...

7 key questions to ask your would-be SaaS vendor

So part of your role in IT is to walk the tightrope. Balance the needs of the business with the realities of IT. As your organization considers online appointment scheduling solutions, your balancing act will also involve vetting potential vendors.

You want to get Janet that short list she's asked for.

What do you look for?

You haven't just spent the last 35 minutes of your life reading this fascinating tome for nothing. In sync with the concepts we've outlined here, your due-diligence process absolutely must include an assessment of how the vendor will allow you to make the SaaS application your own. Here are the 7 key questions that can help you make that assessment:

1. **To what extent can I configure the SaaS application to meet my organizational requirements?**

2. **If I need customization not available through existing preference settings, can you accommodate? How?**

3. **What are the features of the application that enable me to integrate it with my current systems?**

4. **How robust is your API? Can I see your developer documentation?**

5. **How many technical employees do you have who can help with configuration, integration, development requirements, and future rollouts? How do you price those services?**

6. **To what extent do you innovate your SaaS solution? What are your release cycles?**

7. **Can you provide examples of integration and configuration work you've done with other large, multi-location organizations?**

For all the reasons we've just described, the success of the SaaS application within your organization depends on you, and the answers to these 7 questions. (No pressure. ;)) So make sure you're getting complete answers. If you arrive at the end of the list and don't have a firm grasp of the process the vendor plans to use to accommodate your needs, ask.

So, yes — being an enterprise IT professional is a lot like being a tightrope walker. If you completely cave to your business leaders' mile-long wish lists, you're putting yourself and your team on the TGV to the loony bin. But if you refuse to deliver any of the "must-have" features and functionalities your business leaders are asking for, you'll be eating that stale, sprinkle-covered donut for dinner while you search for the nearest unemployment office.

Fortunately, there are ways to make it across that tightrope. Find a best-fit SaaS application offered by a vendor who can clearly answer all your questions, and help you make the case for identifying (and shelving) any "I'd like it, but no one will die without it" features. Once you have buy-in, make the program your own within SaaS best practices:

- Through configuration,
- Through customization within the code base, or
- By building an application on top of the vendor's API.

The business will get what it needs, and you won't OD on ibuprofen.

With pure SaaS, you can have that donut, at breakfast, in the break room.
The way nature intended.

About Kendall E. Matthews

Vice President of Global Marketing and user of "The Force." I've been a growth hacker since 1995, producing like a ZILLION inquiry leads (ok . . . a couple hundred thousand). Also good at taking 2 pm naps.

About AppointmentPlus

AppointmentPlus scheduling software is a configurable, rules-based, pure SaaS product. Our software — combined with our deep product integration knowledge, customer experience focus, and amazing ability to both analyze synergies and synergize analogies — has made us the vendor of choice for many Fortune 500 companies.

Since 2001, AppointmentPlus has been working with businesses to automate their complex customer, staff, and resource scheduling rules. And our Scheduling Cloud™ API has helped more than 215,400 locations worldwide schedule more than 54 million appointments.

Who else wants some awesome-sauce?

See for yourself: Download our Developer's Guide.

References:

1. Gartner, January 13, 2015. "Cloud Service Providers Must Understand Deployment, Adoption and Buyer Complexity to Leverage Cloud Revenue Opportunities."
2. Salesforce.com, "SaaS: Software as a Service."
3. Salesforce.com, "The Seven Standards of Cloud Computing Service Delivery."
4. InformationWeek, "Enterprise Applications: Customization Vs. Configuration."
5. Raj Narayanaswamy, "The Benefits of True SaaS and the Dangers of Cloud Impostors."
6. Workday, "Why We've Moved to Single Codeline Development at Workday."