

Netguru

Handy Guide to SaaS Development

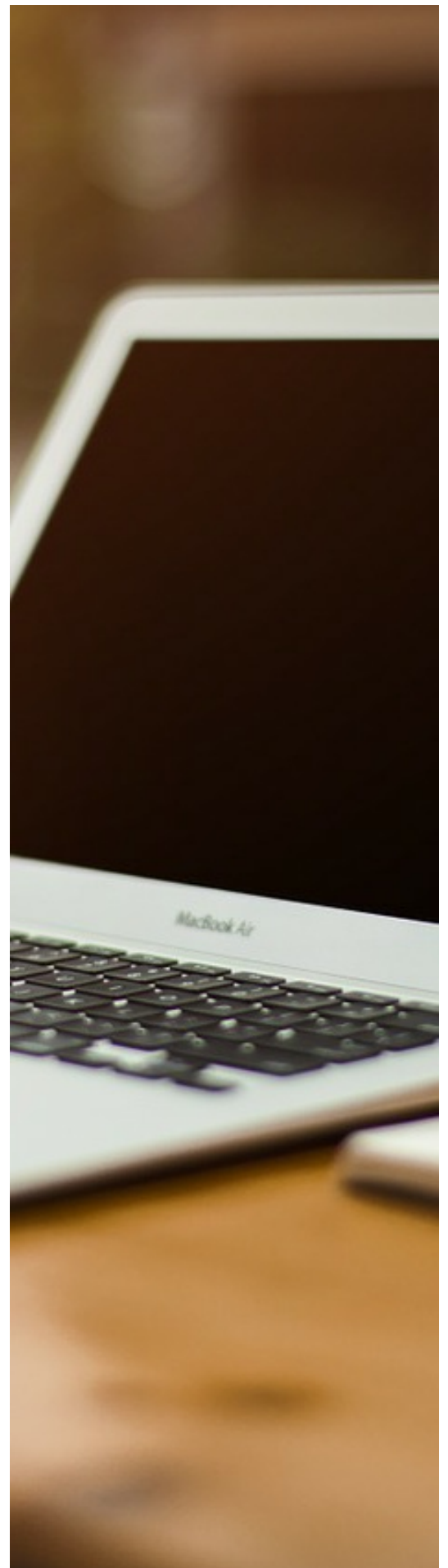


Everything you need to know to create
a great SaaS product

made with
Beacon

Table of Contents

1. How to Build a Great SaaS Product?
2. 7 Most Popular Myths About Creating SaaS Products
3. From Freemium To Premium – What Makes A SaaS Worth Paying For?
4. How To Choose A SaaS Dev Team – The Risks Of Going It Alone
5. How To Choose A SaaS Dev Team - The Traits Of A Perfect Team
6. Summary



How to Build a Great SaaS Product?

Aleksandra Prejs



We found out that many business owners, especially beginners, are in need of recipes for building scalable, high-quality, fast-developing SaaS products. Netguru took the challenge of finding the know-how required to build a great SaaS product – we asked established experts from the industry to share tips and best practices in the field of developing software behind the best SaaS products.

Building SaaS tool – "You need to start with..."

Founders of SaaS businesses often struggle to properly prioritise while developing their tools. And there is a plethora of decisions to be made:

- Should I worry about scalability?
- How much effort should I put into design and user experience?
- Do I "move fast and break things" or rather "observe and improve"?
- How do I know if my product makes sense, what kind of data should I collect?

These are tricky issues which can make or break your idea. We asked some experts who are currently running or investing in established SaaS products, what are the most important aspects of creating a SaaS business. We prepared a short survey with a few questions about creating a SaaS product that really works. Here are the results:

Go fast and lean

First, verify if the core value of your product is in the technology or the business process. If you get this wrong, you will most probably fail. Once you define where the value lies – focus on fast iterations based on closed customer feedback loops. – Marcin Szeląg, Innovation Nest

Most of our experts (almost 90% of participants) believe that the crux of developing a SaaS product is the speed of development and the ability to reach market as soon as possible. **Ship it and ship it fast.**

Take small steps, don't plan too much, observe, measure your KPIs and adapt your strategy. – Paweł Kucharski, CTO at Sotrender.

Pixel perfect execution and great a UX are also very important. Founders think that **implementing analytical tools capable of measuring KPIs is also vital for the whole process.**

Run MVP and LISTEN TO USERS before building advanced functionalities! – Tomasz Henel, TK Investments.

Only one founder disregarded the quality of user experience, giving it a zero on scale of 10, the majority believe you should put focus on this aspect.

Quantify requirements collected from users/customers. Put them in an Excel spreadsheet and keep it up to date. – B. Mozyrko, Usability Tools.

When getting started with your product, **flawless architecture is less important than scalability, stability, quality of code, and proper documentation.** Only two experts decided that it is a key factor at the initial stages of development.

Launch with as few features as possible and learn from your users. – Piotr Kulesza, co-founder of RTA.vc

Keep in mind that the UX is not only regarding the frontend but also the backend of your software. – Brad Kohr, Oracle

Talk with consumers

We also asked the experts about other **important things to consider while developing a SaaS product**. Here is what they shared:

- take the user's perspective into consideration and constantly talk with customers;
- think about the future growth and how the application will work when used by a 100 times more users;
- switch to a growth mindset – when you are flexible you can embrace constant changes and new iterations, bugs in the code won't stop you;
- remember that not all SaaS products are the same, because many aspects of the business depend on the market a given company is trying to address. That's why defining value propositions and having a clear market fit are so essential.
- Treat your startup as a startup, not a full-grown business. You'll get to that point later. Rome wasn't build in a day.

*Understand your customers. Become your own customer. –
Michał Sadowski, CEO, Brand24*

How to kick-off?

So, here you have it – a ready-to-go set of ways to kickstart and continuously improve your SaaS product. Now it's time to learn what mistakes you should avoid while managing your product development process. [How not to run an IT project?](#) We know it.

*Sell before developing. Work on a live organism. – Piotr
Zaniewicz, CEO, Right Hello*

Also, always evaluate if what you're doing is actually crucial for your business? Do you focus on the right things? We'd love to help – and that's what we do during scoping sessions. Check out what it's all about: [How scoping sessions can help you launch a better product?](#)

7 Most Popular Myths About Creating SaaS Products

Many myths have arisen around creating startups and building SaaS products. Nowadays, having your own startup is often thought of as fashionably sophisticated. Yet, there are as many opinions on this as there are SaaS founders. We debunk the most popular misconceptions regarding the process of creating SaaS products.

Myths about SaaS products debunked

Myth 1: It's easy to create a SaaS product

If you truly believe that the process of creating a SaaS product is easy, you could not be more wrong if you tried. No matter what personal qualifications you have, you will always lack some skills and character traits that will be required somewhere along the way. If you are a professional developer, you will need to face some sales and marketing challenges. If you are a business development person, you will still need somebody to write the code for your product. Combining **all the relevant skills together**, looking closely at the timing, designing a marketing plan, reaching the market and taking care of your clients is definitely not a walk in the park, especially when it comes to growing your business.

The biggest misconception is that it's easy. Hardly ever is the initial "product idea" the one that delights the market. More technical teams have a tendency to overcomplicate the product and spend too little time on customer feedback. This leads to writing unnecessary code. – Marcin Szeląg, Innovation Nest

Solution: Before the project kicks off, carefully determine its core values and stick from them at the beginning. If you feel like you could use a hand in this, here's an [infographic](#) and a [blog post](#) to help you out.

Myth 2: Creating a SaaS product is cheap

Let's enumerate the potential costs that you will have to cover at the very beginning of the process: you need to prepare the product architecture, pay developers and designers, research the market, come up with promotional materials and spend money on marketing campaigns. **The idea of your product will inevitably evolve** along with its code, which means constant changes and more work that you will end up having to pay for. While you're at it, don't forget to put a figure on the cost of the time you will spend on this.

Solution: Remember that your product will evolve which means that it doesn't need to have all the features right at the very beginning. Try to estimate the possible costs of launching a MVP and develop this when it starts to make profits or you secure a spot of funding.

Myth 3: Building a SaaS product ends with creating the software

Creating software is only the beginning of the journey towards making a great SaaS product. **Preparing an MVP is the starting point** of the race to attract clients, make software iterations, collect referrals and provide top customer service. In the meantime, you will have to devote time to marketing activities, recruitment meetings, community events and interviews with potential investors.

The launch of your SaaS app is not the finishing line. It is only the beginning of a life-long triathlon. – Piotr Kulesza, RTA.vc

Solution: Focus on you, ability to ship fast and consider hiring proven SaaS developers. Be realistic and don't only estimate the time that needs to be devoted to developing software. Other activities connected with marketing, sales and recruitment are of equal importance, which means you should never underestimate them by consigning them to the category of "I'll do them in the meantime".

Myth 4: I know what is good for my product

No, you don't; your clients do. Of course you're the author of the core idea and you're responsible for its development, but at the end of the day you're not your client. Every bit of functionality you decide to add to your product should be based on your customers' suggestions rather than on your own needs.

Solution: Become your own customer and use your product on a daily basis — **be your own harshest, least forgiving critic**. At the same time, show people what they stand to gain from using your product — this may encourage them to give it a shot, too. Never underestimate a good **user onboarding** process.

Myth 5: My product has to be perfect in every detail

The truth is, your product will never be perfect because **"perfection" is purely subjective**. Some people might call it perfect, while others will merely find it useful and others will think it sucks big time. You just have to face the reality that your product will trigger a lot of different opinions and that's totally normal.

The product must be good enough and delivered fast, then, when you have traction, you can focus on details. — Paweł Kucharski, Sotrender

Solution: Again, specify the must-have features that are absolutely obligatory for **your MVP**. When you build it and put it on the market, you can **hire either** internal or external **developers** and fine tune the details.

Myth 6: Good product sells itself

Not really. Of course, recommendations can be very powerful but to get the ball rolling you need to reach the market first. This means doing a lot of research, preparing loads of curated content, spending time on spreading the word, asking for opinions, tweaking the product, providing top-notch customer service and maintaining your employees' motivation. **A Good product is like a good actor.** There are thousands of good actors and actresses in the world, but only a tiny minority of them are truly famous – those who play in blockbusters with a loads of money spent on promotion.

Solution: Don't care exclusively about your product. Think about all the things around it as well.

Myth 7: Startup is not a business

Yes, it is. Creating a startup means launching a business and then running it no matter if you have 5 or 150 people on board and 1 or 10 products. The problems are the same; the only thing that changes is the scale.

Solution: Be proud of yourself and stay motivated! You're running a company!

From Freemium To Premium – What Makes A SaaS Worth Paying For?

Spotify is one of the biggest players in music streaming market. With over 50 million users, it achieved great global success. Spotify offers 30 days free trial to get you hooked – and they do it well. **30-day free trial is a perfect period to truly discover that music streaming is something that you can no longer live without.**

What Spotify Gets Right, Others Get Wrong

Yes, they do. There are of course many different freemium models. Some offer a free trial that expires after a certain amount of time. Others allow the user to access basic features for free, and then pay for the upgrade (as is Spotify's model, as it happens).

However, **what a lot of startups get wrong is to offer an over-generous free plan.** It's easy to see why this happens – the startup is so excited about its product that it blindly believes that once people try it they will be just as excited and won't be able to hand over their money quick enough. But the reality is that things don't often work out like that – unfortunately we can't all be as big as Spotify, and now we've got it, we don't really need another one.

Why Users Don't Upgrade

There are a number of reasons:

- The free plan doesn't convince users that they actually need the product in their lives.
- The free plan is adequate for most users – upgrading, therefore, seems like a waste of money
- The **onboarding process isn't user-friendly** and doesn't convert many people
- The pricing isn't planned thoroughly enough – that is to say that the focus can often rest too heavily on creating new users, rather than on converting freemium users into paying customers

How To Turn New Users Into Paying Customers

Ok, let's not get too bogged down looking at how some startups do this wrong, and instead focus on how to do it right.

Choose The Free Trial Limits Carefully

A free trial should be both of those things – free and a trial. The idea of this free trial is to get users to appreciate the core value of the product quickly. It's an **onboarding process**, but one that shouldn't last forever. For instance, I've been using the productivity app **Trello** for 2 or 3 years now. I find it so useful in the organisation of my writing schedules, of my colleagues', I can easily collaborate with other users I'm connected with, leave notes, it syncs beautifully between all of my devices, etc., etc., etc. In fact, **the free version is so good that I have never even once made any serious consideration to upgrade** to the paid version. Why would I, when the freebie is so great?

Really, what the people behind Trello should be doing is to give all new users this great experience right from the get-go, guide them through how this fantastic app is going to make their lives easier and more organised – and then take it all away unless they pay for the upgrade. Perhaps Trello might have fewer users this way, but I'd bet my car that they would have more paying customers.






Reward Users For Getting Friends To Sign Up

This is actually a great tactic for encouraging more people to start paying to use your app or other SaaS-style product. And the best way to illustrate this is by looking at a sterling example – Dropbox. As well as paying for additional storage space, Dropbox also lets users earn additional free space by referring friends and family to sign up, or by connecting social media accounts and a few other things (see below image). It's clever because it literally rewards loyal users at the same time as spreading the Dropbox word even further towards potential new users. It's a double incentive to keep using Dropbox.

Earn free space

We've put together some quests and various fun things you can do to earn extra space for free, such as:

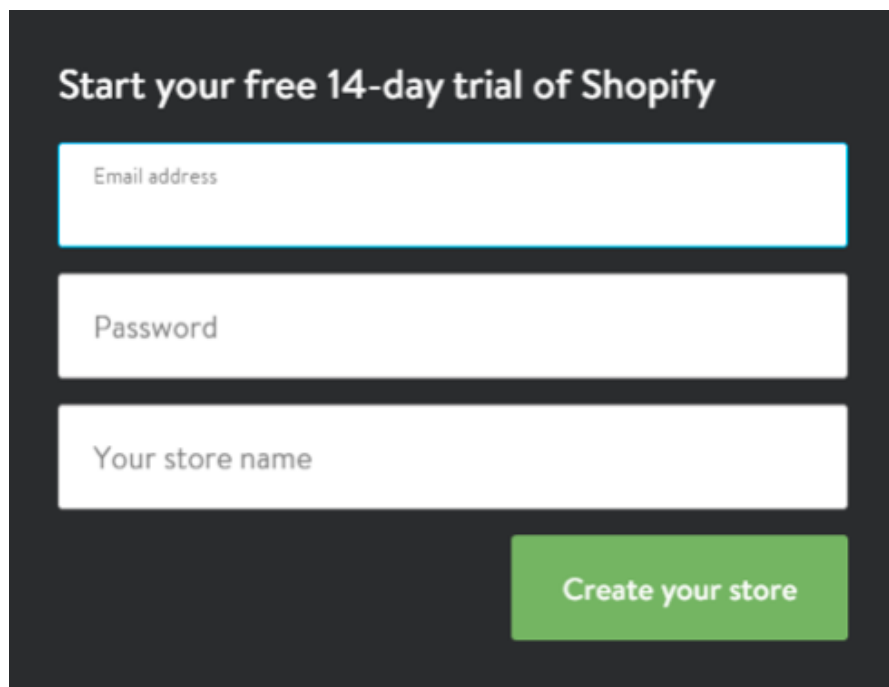
- Refer your friends, family, and coworkers to Dropbox
 - Please note that sharing a folder does not count as a referral. To refer friends you'll need to sign in to your account, click **Settings**, and then click "invite your friends."
- Complete our "Get Started" guide

	Get started with Dropbox Take a tour of the basics of Dropbox	250 MB
	Connect your Facebook account Share folders with your friends and family in a snap	125 MB
	Connect your Twitter account Invite your friends to Dropbox with a tweet	125 MB
	Follow Dropbox on Twitter Stay up to date with the latest Dropbox tweets	125 MB
	Tell us why you love Dropbox We'd love to hear your feedback	125 MB

See all of those ways that you can start earning free space? It's great. **Dropbox makes you fall in love with their product before they start asking for money.** Eventually, the free space will run out, but by this time we value Dropbox enough to start paying for it (yes, I'm a premium customer of Dropbox, by the way – I wonder if I'll get any freebies for bigging them up.

Make The Initial Sign Up Process Easy

Keeping things simple is crucial for to the initial conversion, the onboarding and finally the upgrade process. However, as you can see there are three clear steps that need to be taken here, and the last thing that you want to do is create an obstacle at the first one. Indeed, in the first instance, **when trying to get users to fall in love with your product, you need to keep things as simple as possible every step of the way** – and that includes the sign-up. Remember, when all is said and done, you will have created your product to solve a problem, or to put it another way, to make something less difficult or simpler than it currently is. Users place value on simplicity, so showcase that you're on the user's side right from the start, as in this example from [Shopify](#):



The image shows a sign-up form for a 14-day free trial of Shopify. The form is set against a dark background and contains three white input fields stacked vertically. The first field is labeled 'Email address', the second 'Password', and the third 'Your store name'. To the right of these fields is a green button with the text 'Create your store' in white.

Getting User To Pay For Added Convenience

One tactic to convert that freemium user into a premium one is to make convenience the very thing that they need to pay for. Let's take my Spotify example again. Although still a great music player, I have to say that the only way to get the full brilliance of Spotify is to use the premium version. With the free mode, you can only shuffle play, have to put up with adverts, can only listen to music when you're online – it's like a completely different platform to be honest. But, the transformation upon upgrading is phenomenal. Now you can download music to listen to wherever you are, no matter if there's an internet connection or not. No adverts. You can play any song you like – it's more than convenient. It's pure musical bliss.

What's the difference between **Spotify** and **Spotify Premium**?

	MOBILE FREE	DESKTOP & TABLET FREE	PREMIUM
Shuffle play any playlist or artists' catalogue	✓	✓	✓
Create your own playlists	✓	✓	✓
Play any song on-demand		✓	✓
Highest quality audio			✓
Uninterrupted listening			✓
Download music and listen offline			✓



Consider "Cheapium"

There is a problem with freemium plans – they don't always pay off. There's a [brilliant blog](#) by the Hubstaff team, who go into great detail about how they learned this the hard way, but the key points can be boiled down to the following:

- Paid products carry more value
- Free users bring more free users
- Free users eat up support
- People take advantage of free accounts

The lesson that Hubstaff learned is that if **someone values a product, they will pay for it**. And so they ditched the freemium model in place of a "cheapium" one. **When people are paying for something, they tend to respect it more**. Perhaps it's human nature, or a sign of the spoilt times that we live in, but it's true. By going for a cheapium model, where users have to pay just a small amount for basic features, you can almost be sure that they will pay a lot more attention to the product – and that might just be the key to them discovering just how brilliant your app is. It might seem counterintuitive, but we value the things that we have to pay for more than the ones that we don't. Exploit that human trait.

Make The Free Trial Short (Usually)

The 30-day free trial of Spotify worked because of the gigantic product that Spotify is. However, this is month-long period is the exception to the rule, for usually, the best option is to keep the free trial period short and sweet – 14 days at most.

People don't tend to take full advantage of a 30-day free trial. It's too long. The [Close.io blog](#) says that the vast majority of users trial a product for a day or two, and then either sign up or don't come back. And this is why you **need to strike while the iron's hot and keep that free trial period short**. When people feel like they have less time to trial your product, they are more likely to actually take the time to get to know it quickly. When we know we've got a whole month to do something, it's pretty easy to put it off, and then forget about it altogether.

Are Your Free Trials Short Enough?

Long trial periods (more than 14 days) usually do more harm than good. Because most people won't even try your product for more than 3 days in a row. They'll be more engaged with your product if the trial is short. And your conversion rates will go up.

If your clients need more than 14 days to evaluate your product, **make them ask for it.**

You will reduce the number of free trial signups, but the signups will be of higher quality. And **focusing on high-quality leads will generate more value for your startup.**

Most importantly: **Try different trial length periods** and how they affect your conversation rates. Start with a 14 day trial and compare how it performs against other trial lengths.

B2B Startup Sales Advice - Close.io

(Image source: [Close.io blog](#))

How To Choose A SaaS Dev Team – The Risks Of Going It Alone

The cloud has been – and will continue to be – one of the greatest assets to the modern business, and particularly SMEs. With monthly pay-as-you-go pricing schemes lowering barriers to entry for even the smallest of businesses, combined with flexibility, scalability and accessibility, it's no wonder that the future will skew increasingly towards SaaS models as time goes on. But there's a problem. You know that you're not going to be able to build, deploy and maintain the product all by yourself – so how do you go about choosing the perfect SaaS development team that's going to turn your idea into a very lucrative reality?

SaaS exist in a permanent state of beta, which, far from being a bad thing, simply means that **users of the software can expect an almost constant stream of updates and upgrades** – all included within the same monthly (or annual) fee – for as long as they are customers. It certainly beats licensing models, where people had to fork out for expensive upgrades every couple of years – or not, if it was simply beyond the SME's price range.

You may have dabbled in some development in the past – perhaps built a few apps, or what have you – and now you think you've got the perfect idea for the next big SaaS product. Learn why you shouldn't do it all by yourself.

The Problems In Taking On the Whole Project Yourself

Of course, you may harbour the idea that, since you already have some experience in coding, you're going to be able to take on the whole project all by yourself. However, there are some things that you need to take into consideration before embarking on such an endeavour.

These are, in no particular order of importance:

How Can You Be Sure That Your Code Quality Is Good Enough?

SaaS is a very specific type of IT product, which requires expert knowledge from whoever is behind the development, as well as special qualities in the workflow. While it's most certainly true that **the success or failure of your project won't rest entirely on how well-built your software is** (SaaS means permanent beta, remember – which means you will always be able to update and improve), it nonetheless needs to be robust and pretty much bug-free if people are going to actually adopt and use this thing. In many ways, so long as your SaaS is solving a genuine problem, and your vision is near enough met during the build, then you will have the basis for a very successful product. However, people don't buy into ideas – i.e. **if you want to be successful, then you're probably going to need stronger coding skills** than what you can acquire from reading a few blogs, books, or taking an online crash course.

How Will You Handle Testing?

One of the fundamental aspects of building any software is extensive testing. Many development houses these days go in for an Agile development approach, where multiple testing is conducted within each **iteration** of the build. This is an extremely effective way of building software, for it means that various additional functionalities can be added in relatively short time to the software product. But how and where are you going to test all of these mini-updates that you build and release? **You will need to know how to deal with bugs that you find within each iteration**, and, doing all of this alone is not simple.

How Can You Be Sure That Just One Developer Will Be Enough?

You can't. In fact, there are probably very few occasions where just one person can accomplish the whole build – let alone the marketing and the maintenance – alone. Besides, if you're just doing it yourself, there's no one there to challenge your ideas or test your code. To be frank, it spells disaster. Far better is for you to **establish yourself as the technical founder, but then to go into a partnership with someone whose purpose is to market and sell** what you have produced, and then outsource a team (and it can be a small team) of professional developers who know exactly what they're doing.

Indeed, what you have to think about is the fact that, if you are determined to tackle this task by yourself, you are essentially going to have to be founder, project manager, developer, marketer, salesperson, client service and quality assurance all in one fell swoop. Be honest – can you really handle all of this responsibility alone?

Which Programming Language To Use? And Only One?

The likelihood is that when you begin to develop your SaaS product you will be able to rely on just the one programming language and one framework to begin with. And that might be ok for a while – and of course you might not have any other choice if you're only versed in a single language. But, what happens if it turns out that you need to learn more languages to achieve what you want for your product? Does the whole project get put on hold whilst you take yourself back to school to get to grips with code that is completely new to you? If so, then surely you're going to need to build a few basic programmes first whilst you're learning, then gradually work your way up to more advanced constructions. How long's that going to take? Six months? A year? Longer?

Quite possibly.

These are all questions that Doug Keefe goes at length to answer in his brilliant LinkedIn post '[What Language Should I Use For My SaaS App?](#)' I'd encourage all those who are considering developing a SaaS product to read the whole thing – **whether you're planning on hiring in a development team or going it alone – but the key take away from it is that there is no silver bullet.** What you want your product to do will probably be achievable in many different languages – but do you have the knowledge to make the right choice, and the skills to move from one technology to another should the need arise?

How To Choose A SaaS Dev Team - The Traits Of A Perfect Team

We explained the risks and pitfalls of enthusiastic entrepreneurial coders taking on a whole SaaS project alone – it might be somewhat irresponsible and prone to failure. Now, let's move on to the building and running of a rocking SaaS product by a team. What kind of team should you look for?

What You Should Be Looking For In The Perfect SaaS Development Team

Code Review

If you decide to build your SaaS product all by yourself, then you can never really be sure that your coding is robust enough during construction. Sure, you may be able to hash together a workable programme, but, without other people checking and testing your code on a regular basis, you're more likely to let yourself "get away" with taking a few shortcuts, or turning a blind eye to some glaring flaws that perhaps a few colleagues would be more willing to pick up on.

In short, **you need to assemble a team whose members all have the same goal in mind** – to create a durable, robust code structure that is going to deliver to consumers exactly what it says on the tin. For the same reasons that writers need editors, coders need fellow experts to pick over what they've done and pick out the bits that need improvement. It can be incredibly difficult to spot the flaws in your own work – but a dedicated team will help each other through.

Cooperation and Communication

The key to success in almost every endeavour is of course communication. If you're working alone, you're going to be spending a lot of time talking to yourself – and if you're the one asking the questions, then you'll probably find yourself a little stuck when expecting answers.

A great development team, on the other hand, will be a communicative team. It will be a team full of people who ask each other questions and keep each other – and the client (if it's not you) – up to date precisely with the progress of each iteration of the ongoing project. All members will be proficient in and happy to use [project management apps](#), and will know exactly [what it takes to run a successful IT project](#). Put simply, **your development team will be willing and able to collaborate and communicate with everyone involved with the project from the get-go**, and will indeed share your philosophy that communication is everything when it comes to success.

Proficient In Online Payment Systems And Processes

You will of course be undertaking the massive endeavour of creating a SaaS product with profits in mind – and you need to pick a team that understands that. At least one member of the coding squad should be proficient in building in extremely robust and, above all else, secure payment gateways so that they can put it into your finished product. What is more, **you need to make sure that this person (or persons) understands the differences in payment options**, and, as such, will be able to advise and discuss with you what is most suitable to the SaaS that you are constructing. Will a freemium model be best, for instance, or do you want to build a straight up premium product? [Cheapium](#) is another option that is gaining in popularity – and if you haven't heard of it, or the reasons why some experts are arguing that it's a better approach than freemium, then this all the more reason why you need an experienced person on your team who is fit to advise you.

The Ability To Plan And Explain Work To The Client

If you're heading up the development of a SaaS product on behalf of someone else, then you need to make sure that your team is able to communicate to that client in appropriate terms. Business people are nearly always proficient in communicating their ideas, but programmers are not always proficient in communicating complicated processes in layman's terms. This, is a project management problem that you will have to overcome, and it will involve dividing your project plan into bite-sized chunks, creating lots of visual mock-ups and graphs, and engendering a sophisticated communication schedule that all members will rigorously adhere to. The whole process needs a blog post all of its own, and it just so happens that we've already put one together, and so I refer you to '[How To Communicate Your Ideas To Developers](#)' for more advice and information.

The Ability To Think, Work And Act Rationally

Development has no room for divas. Well, the world in general doesn't but you know what I mean. Your team should have the ability to think, work and act rationally – as a team. This means that **there can't be even a single person who in any way, shape or form takes the "I know best" attitude.** Problems will arise, and challenges will have to be overcome – that's all part and parcel of building great SaaS. What you don't want is a bad apple making things difficult for the communication of the team. **Trust your intuition when hiring** and weasel out any divas or rock stars before you start to build – the success of your SaaS project depends on it.

An Understanding Of The SaaS Architecture And Security Measures

SaaS is a very particular type of IT product. It requires a team with a very specialised set of skills, and a very clear understanding of the security measures that need to be taken to ensure successful and safe deployment to multiple subscribers. Additionally, SaaS architecture means that a fast, secure and easy-to-use data services layer needs to be developed – especially if you're going to need to develop a public API. Indeed, **the success of your SaaS product will probably mean that you are going to have to allow for its integration with other software packages, and/or allow for the development of additional capabilities by third party developers, as well as providing services for mobile applications – and all of this means consistent and dependable APIs.**

The Skills And Expertise To Develop Mobile Apps

In 2016, the world is mobile – that's not going to change. You or your client first conceived of the idea of this great SaaS product for desktops, there will nonetheless come a time when your subscribers will simply expect to be able to access the product on their smartphones and tablets as well. When it comes to SaaS, the case is that **many projects will require native apps developed separately** to accommodate these devices – and so it makes the best sense to appoint a team which contains members with the skills and expertise to develop mobile apps from the start, rather than make your clients wait until you build the mobile apps later. You will risk losing your subscribers to competitors if you go for this approach.

The current limits that HTML5 imposes on mobile applications means that creating a viable user experience (UX) on mobile is presently not best practice, as [David Key](#) from Cloud Strategies explains:

“The majority of SaaS vendors will provide both a desktop and mobile application. While HTML5 is rapidly become de rigor on desktop, its use on mobile platforms is situational based on the application requirements. According to [Flurry](#) (April 2013), only 20% of the activity on mobile devices is in HTML browser (though B2B applications would likely be different).”

Have Understanding And Experience In Big Data Collections And Management Systems

The collection and management of client data is no walk in the park – and, when the new regulations from the EU officially kick in over the next couple of years, the seriousness of handling people’s personal data cannot be overstated.

The [EU’s General Data Protection Regulation \(GDPR\)](#) means that hefty fines are now in place to ensure that data protection rules are taken extremely seriously by all companies who obtain data on EU citizens – whether your business actually resides in the EU or not. A company can now be fined up to 2% of its global annual turnover just for not having its records in order or by failing to conduct risk assessments on the data in its control.

Furthermore, 4% fines can be issued for more serious matters, such as violations to data security and the conditions of consumer consent for having their data collected and shared. From a development perspective, the most critical point about the GDPR is now that privacy precautions must be built in by design and default – not merely as an afterthought. This means that **your developers have to be building data protection facilities into your SaaS product right from the word go**, and you must be able to provide clear and unambiguous evidence of this. No shortcuts can be taken, no stones left unturned. Users must also give clear and unambiguous consent for your SaaS to collect their data, which must be indicated by an action (such as the ticking of a tickbox). Consent, cannot be implied any longer – i.e. just because people are signing up to your product does not mean that you can assume that they wish you to collect their data. Consent must be actually given, not implied. Needless to say this is serious stuff, and your development team needs to understand this and ideally have experience in developing software that incorporates data protection.

Be Ready For Frequent, Non-disruptive Updates

Consistent, uninterrupted uptime is of course one of the key aspects of SaaS – indeed, if you can't deliver this, then you're not cut out for SaaS. **One of the beauties of SaaS is that frequent updates and innovations to the product can be delivered to the user, who will always be enjoying the most up-to-date version of the product whenever they log on to it.**

This is no mean feat for your developers, who must know how to plan and execute these updates, all without disrupting the continuity of the service. This means that a fault tolerant architecture must be in place under the hood, with robust fail-over plans in place in case difficulties occur. Put simply, **the software must be architected to anticipate and recover from failures** without any (or only minimal) impact to the service.

SaaS is difficult and demanding, there's no denying that. Ask yourself: is your development team capable of providing all of these things? If you need a helping hand, don't hesitate to get in touch – we've got the experts that can provide all of the above and a lot more. We'd love to hear your idea for your next project!



NOW YOU KNOW EVERYTHING

Got an idea for your product? We'd love to help!

With over eight years of experience, our team of 200 dedicated individuals have created hundreds of successful SaaS products. Click the button to learn more:

I WANT TO CREATE A SAAS
PRODUCT

made with
Beacon