INCRED

FIND YOUR WAY TO SUCCESS

Enterprise SaaS Development Guide - Kickstart Your Software Business

Table of Contents

Introduction To SaaS Development

SaaS is all the talk these days. It's taking over everything. Thanks to the revolutionary software distribution model, businesses are now able to save a ton of time, money, and human resources at remarkable levels. SaaS solutions not only cost around three times less than traditional software solutions, but they're also way more flexible, scalable, and easier to use.

If you want your business to thrive in the years to come, it's in your best interest to take advantage of the paradigm shift that is currently going on. For one, SaaS has radically changed the way companies are managing, storing, and exchanging data. But you'd best buckle up because it's going to be a bumpy ride!

To get a better appreciation of SaaS and have a better perspective on what's to come, it'd be best to get yourself familiar with how the business model evolved over the past decades.

Evolution of SaaS

Sometimes used interchangeably with cloud computing, the term "SaaS" or "Software as a Service" was first used in 2011. The business model that gave birth to it goes way back -- about 60 years ago when IBM offered "time-sharing" to its customers. If you have no idea what that is, think of typical keyboards and monitors attached to a main computer. It wasn't strictly "Software as a Service," but by using a shared resource environment, it was groundbreaking at the time and it sure as hell beat going through massive paper ledgers every time!

Soon, the dot-com boom came, and by the end of 2001, it had become apparent that the Internet was here to stay. As the Internet became widespread and more embedded in our daily lives, the computing industry implemented an early form of SaaS called ASP (Application Service Providers), a business model offering IT-enabled solutions to customers over the web. It's also worth mentioning that during the ASP set up, each client was required to install the software locally on their PCs.

ASPs, however, didn't have the infrastructure to support "multi-tenancy," meaning you needed to maintain a separate instance for each client. This, of course, took up a lot of resources and undoubtedly created headaches!

Unable to scale and handle the oncoming barrage of client requests, ASPs became riddled with performance issues. Soon enough, a number of ASP companies such as Industry Consortium and FutureLink Distribution Corp began shutting down left and right.

While some ASP companies were able to weather the storm, the age of SaaS 1.0 was beginning to make its mark on the computing industry, led by none other than Salesforce.

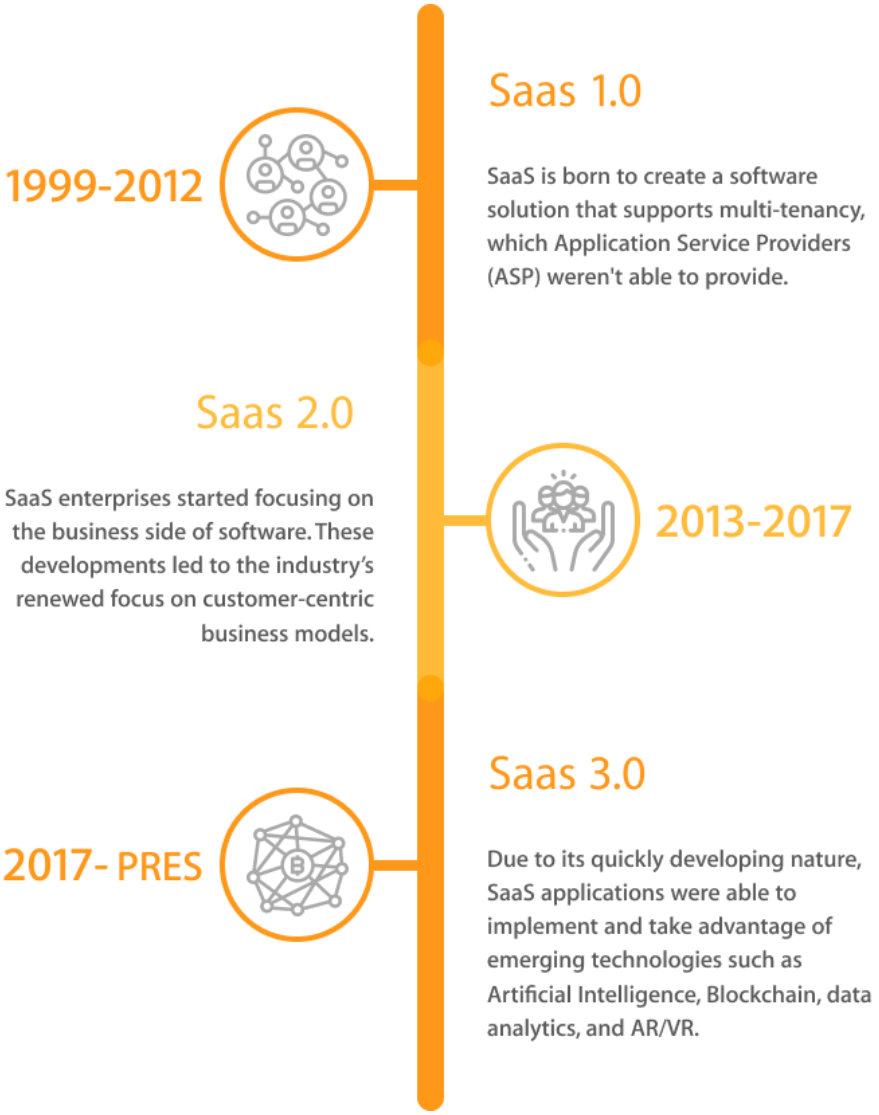
Founded in March 1999 by three entrepreneurs, Salesforce was formed to provide SaaS solutions specifically. Though most of its revenue comes from its CRM (customer relationship management) product, Salesforce also offers consumers a variety of commercial applications on the web.

The idea behind Salesforce was as simple as it was revolutionary: to deliver software exclusively via the web using a subscription-based model. No installations, no upgrades, no nothing. While other computing companies like Zendesk and Slack followed suit, Salesforce maintained its dominance in the SaaS space. In fact, the company's Q4 numbers indicate that its market cap has soared to $88 billion.

SaaS 2.0 came onto the scene in 2013 when enterprises started focusing on the business side of software. These developments led to the industry's

renewed focus on customer-centric business models. At SaaS Fest 2016, David Cancel of Drift said in his talk that SaaS 2.0 will be ruled by the customer, adding that companies who put the needs of their customers first will dominate the competition.

This is evident today. In our increasingly digitized society, customers are expecting more, challenging businesses to deliver a seamless, omnichannel experience while using the best technology to provide personalized services.



SaaS 3.0 Development Technologies

Due to its robust infrastructure, exceptional scalability and seamless integration with third-party services and tools, SaaS applications were able to implement and take advantage of emerging technologies such as Artificial Intelligence, Blockchain, data analytics, and AR/VR (Augmented Reality/Virtual Reality). This signaled the start of the SaaS 3.0 era.

Artificial Intelligence, in particular, has often been dubbed as “the next big thing” in the technology industry. With SaaS serving as the defacto delivery model, a handful of AI-infused apps are getting smarter and more intelligent than ever.

Big players like Amazon, Oracle, and AWS have been carving out a place for themselves in the AI space by developing “AI as a service” cloud tools. However, as cloud services and enterprise technology are becoming cheaper, small to medium-sized businesses are starting to catch up to their larger counterparts. This is even more apparent because anyone with extensive knowledge of the five primary programming languages (Python, Prolog, JAVA, C++, and LISP) is capable of building AI technologies that have the potential to disrupt the market.

AR/VR technology is also picking up speed and covering a lot of ground thanks to the SaaS model. With SaaS authoring tools like Lectora Online, Gomo, and dominKnow available online, you can easily create 360VR interactive environments without entering a single line of code!

The AV/VR talent market is also growing. As is the case with AI technologies, anyone with a knack for programming languages such as C#, Java, JavaScript, Python, and Swift can get in on the AV/VR scene as a developer.

The Blockchain sector has also significantly impacted SaaS. As the number of people who want to build on Blockchain continues to increase, so does the demand for SaaS solutions tailored to Blockchain technologies.

What's more, with intuitive APIs being implemented across main programming languages like C++, Python, Ruby, Go, and JavaScript, the development, and deployment of Blockchain apps are becoming easier and faster than ever.

So what does the future hold for SaaS as a business model? One thing's obvious, SaaS is not going anywhere soon. According to a 2017 study conducted by BetterCloud, 73% of enterprises will run almost entirely on SaaS by 2020. And you'd better get in on the action if you want your business to thrive and survive through the massive changes that are ahead of us.

Preparing for SaaS Development

Chapter 1

Preparing for a SaaS development project can be a daunting task for most project managers. For one thing, building a SaaS enterprise solution is a process with many moving parts. If you're not careful, chances are your project won't see the light of day outside of the development phase.

Like every process, SaaS enterprise development requires specific steps and methodologies, not to mention the expertise, to bring a successful product to market. Every big project needs a strong foundation to build upon. By laying out the initial groundwork for your SaaS project, you can create a solid base upon which everything else follows and grows.

To put it simply, if you want to get everything right, you have to make the correct preparations. Here are four essential steps to ensure that your SaaS enterprise project becomes a success.

User Research

Your SaaS product is bound for failure if you don't take the time to understand your target user. If you don't understand their pain points, it's highly unlikely that you'll be able to provide them with real solutions that they'll need. To learn more about your target user, you need to do some extensive research. You can begin by asking the following questions:

- *What features are they looking for?*
- *What are their pain points?*
- *What can your SaaS do to address these pain points?*

You have to do a whole lot better than just answering these questions in a broad manner. For your SaaS product to be a big hit with your consumers, you have to narrow down your user base and understand your target audience all the way down to the most minute details. One of the most effective ways of doing so is by creating user personas.

User personas are character sketches that represent your prospective users, and in the case of B2B marketing, decision-makers. The more detailed your user persona is, the better it will serve you in terms of propelling the growth of your SaaS application. What's more, a detailed user persona gives your marketers something to work with when it comes to building your brand's authority and engaging your target audience.

If you don't feel that user personas are essential to your business, bear in mind that a 2016 study by Cintell reported that 71% of companies that exceeded their revenue and lead goals used documented personas to drive their marketing efforts.

Interviews

If you've already developed a SaaS product before, you can interview ideal users from your existing customer base. You can ask them the following:

- *Details about their personal background (age, annual household income, place of residence, educational background, number of children, etc.)*
- *What is your job role? What tools do you use in your job?*
- *What are your biggest challenges in business?*
- *How do you define success in your role?.*

Tailor your questions to your specific business needs so you'll get the answers that'll help you better target the needs of your user base. For a comprehensive list of questions to ask your target users, you can check out 33 Degrees Design Studio's free template.

Demographics and interest reports from Google Analytics

Another useful method that can help you add more detail to your user persona is

to use Google Analytics. Analytics reports, after all, provide comprehensive data that tell a great deal about the behavior of users when they use your app. By drawing insights from such data, you can determine what users want to achieve. Knowing what they want to achieve will help you come up with ideas or solutions that can help them out.

When you go to the *Audience* section and select *Demographics*, you'll see a detailed breakdown of your site's visitors, including their age and gender.

Go to the *Key Metrics* drop-down menu and you can change the metrics to:

- *% new sessions*
- *Average session duration*
- *Bounce rate*
- *Pages/Session*
- *Sessions*

By looking at and analyzing each of these metrics and exploring their correlations between different demographic segments, you can come up with actionable insights that will help you address the needs of your target audience.

Google Analytics' *Interest Reports,* as the name implies, gives you detailed information about your users' interests. Go to *Interest Reports* and then go to *Audience* then *Interests*. Once there, you'll see three main categories, which are:

1 **Affinity Categories:**

These are broad content categories that provide insights about your users' lifestyles

2 **In-Market Segments:**

This particular section shows users who are looking to buy specific types of products and services.

3 **Other Categories:**

This segment shows a more focused view of what type of products and services users are interested in buying.

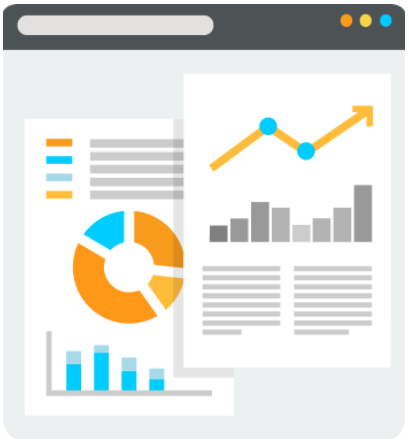
Social Listening

If you're just starting out, you can enrich your user persona by looking at your competitors' customers. After all, they're a part of your target audience too. Tracking social media mentions of your competitors can help you acquire key insights that can inform your target audience's purchasing behavior. There's a ton of social listening tools that can help you do just that, including HootSuite, BuzzSumo, Tweetdeck, Google Alerts, and more.

Once you have a fully developed user persona, you can use it as a guide that will inform your decisions as you build your SaaS app's features. For example, if customer churn is one of your prospects' major pain points, you can develop a feature that makes it easier for them to personalize their interactions with their clients.

If your SaaS app caters to businesses, keep in mind that business decision makers have a different purchasing behavior from that of regular consumers. C level execs tasked to look for SaaS solutions are likely to spend more hours online before deciding. Because they are accountable to others within their organization, they are going to look closely at the particular needs of their business before picking a SaaS solution. They may also need to consult with other decision makers within their respective organization, each with a different area of expertise.

Market and Competitor Research



If you want to get a competitive edge in your industry, you need to know what you're up against. Therefore, it's imperative that you conduct extensive market and competitive research before you release your SaaS enterprise app. Your competitors, after all, can directly impact the sales of your own products. More importantly, the only way you can provide something unique to your prospects is to know first what's being offered by your competition.

Take your competitor research further by making a list of direct competitors. Ask yourself the following questions:

- *What are the products and services they offer?*
- *Are their products and services aimed towards satisfying the needs of your target audience?*
- *What marketing strategies are they using?*
- *What is their pricing structure?*
- *What is their total sales revenue?*
- *What's their percentage in the overall market share?*

Once you've worked out the answers to these questions, you can start brainstorming with your partners on how you can differentiate your brand from your competitors. Do you think you can offer a similar product at a lower price? Are there underserved marketing channels you can capitalize on? From there, you can come up with a unique value proposition that will convince your prospects that your product or service is the better option.

It pays to know which software engineering languages your competitors are using for their respective SaaS apps. Monitoring the relevant details of how they

implement their products can reveal a great deal about processes and tools that you're missing out on. It helps to identify specific strategies your competitors are using. Perhaps you can adopt these strategies yourself to better address the specific needs of your business and your customers.

Speaking of tools, BuiltWith is one competitor analysis solutions provider that can help you "spy" on your competitors, particularly the technologies they are using. The tool allows you to check out the software engineering languages they're using for their web applications. Even better, it provides access to market analysis data that will help you edge out your competitors.

Using it is easy. Just go to BuiltWith.com, enter the URL of a competitor in the *Lookup* field, and then press *Enter*. Almost instantly, you'll get a full list of technologies that said competitor is using, from analytics tools and widgets to mobile apps and Javascript libraries.

MVP Planning

- *It enables you to gain valuable feedback that will help you improve your SaaS app in its developmental stages.*
- *It tests and confirms the initial assumptions you've made in the pre-developmental stage.*
- *It allows you to determine if your software solution, even in its barest essentials, will address your customers' problems.*

An excellent way to pull this off is to implement an MVP process. For those who are not familiar with the term, MVP stands for Minimum Viable Product. As the

name itself suggests, MVP is an early version of your SaaS app. What it essentially does is help you get to the full version of the product within a shorter time frame.

For the MVP process to work, start with a prototype that aims to solve one major problem for the customer. The goal here is to give early adopters the opportunity to validate their assumptions about your product's usability, enabling them to come up with insights and suggestions that would help you improve your product before its official release.

Jumpstart your SaaS development with Rapid Prototyping

Conceptualizing a product or service is one thing; releasing it to the world is quite another. Developing a SaaS solution is a long, complex process and integral aspects of communication may get lost in translation. This is where rapid prototyping can come in handy.

Rapid prototyping in SaaS development is a process where the team creates a mock-up version of your product's final version. In short, you're creating an incomplete copy of your product but with its most basic elements and functions intact. With rapid prototyping, you and your team can get a better grasp of what the final version will look like and how it will function with minimal time investment. It helps you test ideas in regards to its overall design and functionality. Additionally, it can also help the development team come up with insights that will help them refine the product further towards the project's completion.

Types of Rapid Prototypes

Low-Fidelity (lo-fi) Prototypes

Low-fidelity prototypes contain only the most basic elements of the final product, including visual elements, key content, structural hierarchies, and more. Creating a lo-fi prototype usually takes less than an hour, allowing the development team to explore ideas and hypotheses in a shorter time frame. It's also affordable, making it the more attractive choice for development teams who are working on a tight budget.

One major downside of using a low-fidelity prototype is that its bare bones approach may result in inaccurate testing. Due to its limited interactivity, interaction flows may not be as detailed, and things may not translate well in the implementation stage.

High-Fidelity (hi-fi) Prototypes

A Hi-fi prototype is an interactive prototype that looks and functions like the final product. Since its visual elements and content are more detailed, high-fidelity prototypes often result in more accurate and effective testing. It's highly interactive, allowing the team to think more critically about technical considerations they may face in both the developmental and implementation stages.

Unlike lo-fi prototypes, hi-fi prototypes are more expensive and require more time to create. A project that has a limited budget may find it difficult to convince stakeholders to throw their support for the product.

Choosing the Software Engineering Language

Chapter 2

At this point, you already know that your enterprise SaaS solution makes sense -- both from a marketing and user experience standpoint. Your next order of business is to come up with a structured roadmap outlining the features and functionalities that you need to implement in the forthcoming releases. Make sure that you make the necessary tweaks and polishes to ensure that all issues that were raised in the MVP phase have been resolved.

With that part done, let's get down to the nitty-gritty -- choosing the software engineering language that will be used to develop your SaaS app. You want to make sure that the technologies used for the project are efficient, fast, robust, and conducive to iterations.

Those who don't have a deep background in programming may find it difficult to wrap their heads around the technical processes involved. In this case, it's imperative to know that web or software development is composed of two parts: the front-end and the back-end. The front-end side of development is often referred to as the client-side while the back-end is often called the server-side.

Front-end

Front-end or the client-side is the part of the website or the app that is visible to users or clients. In other words, the front-end constitutes the user interface (UI). As you can probably tell by now, front-end developers are mainly concerned with the visual aspects of the application. They use programming languages and frameworks to ensure that websites work properly regardless of the device used.

Front-end developers typically use CSS, HTML, and Javascript to create the site and web apps that users interact with.

Back-end

Back-end or the server-side typically refers to the guts of the app that lives on the server. The back-end is hidden from users, but without it, the websites and the applications they are using will merely cease to exist. The back-end, after all, is where the heavy calculations and the data wrangling are done. In other words, the front-end or UI owes the front-end a boatload for doing most of the dirty work.

Back-end developers use a wide range of programming languages and frameworks, including PHP, Python, Java, and Ruby, not to mention database systems such as Oracle, MySQL, and SQL Server.

Next, you'd want to pick the software engineering languages that will be used in developing your SaaS application. At the very least, you should have minimum background knowledge of different programming languages. That way, you'll be in a better position to check whether your business objectives are being met.

There are two possible scenarios when it comes to hiring a team of developers. Either you choose to hire a CTO (chief technology officer) beforehand or work with a third party agency. If you go with hiring one, chances are the CTO already has a programming language in mind. Since the CTO is responsible for overseeing

all technical aspects of the SaaS project, you don't have to do much beyond communicating your startup's strategic goals. The CTO should be able to carry these goals out while communicating and collaborating with the entire development team.

If you're still looking for a CTO, then you'd need to have at least a general idea about which programming language will be ideal for your SaaS enterprise app before you start screening candidates. It will end up saving you more time in the end.

Choosing a software programming language is where it gets really tricky. For one thing, there is no one-size-fits-all when it comes to picking one. But you can narrow down your list by using the following criteria in the selection process.

Top 5 Criteria in Choosing a Programming Language

1 Database Types

Every software deals with data on a regular basis. But if it doesn't have enough resources to manage and process data efficiently, your users will end up frustrated. So for better results, pick a programming language that provides an excellent environment for database-building. That said, the best programming language for your SaaS solution will depend mainly on the type of database you want to use. The following are different database types you need to keep your eye on.

- **Relational Database**
  Is your SaaS application more focused on data integrity? Is it meant to handle highly-structured data? In such cases, relational database management systems like Oracle, MySQL, MS Server, or PostgreSQL are ideal.

- **Document Database**
  If your software is geared towards managing semi or unstructured data and processing in-depth data analysis, you're better off using document database systems like MongoDB and Couchbase.

- **Wide-column Database**
  Wide-column stores, on the other hand, are ideal for apps that require significant data analytics and large-scale projects on account of their processing speed. Cassandra, HBase, and Accumolo are known to be proficient in handling large datasets.

- **Key-value Database**
  SaaS apps built for optimizing data management at scale will work best using key-value database systems or key-value stores. Redis and Memcached are currently the most popular key-value stores in the market.

2 Data transmission

Speed is of the essence if you want your SaaS enterprise solution to deliver in terms of user experience. As such, you're better off picking a programming language that delivers a reliable and robust network data transmission. The faster the transmission speed, the better the performance will be.

3 Early prototype testing

A good programming language should leave a lot of room for early prototyping. The earlier you can test a software's usability, the earlier you can get user feedback. In the end, it saves you more time and money, allowing you to make the necessary adjustments and modifications before heading into the MVP process.

4 Speed and performance

The fastest and best performing programming language will depend on your field. But for general purposes, Python or Ruby is ideal.

5 Potential for scalability

Scalability is integral to a successful SaaS platform. After all, your business can't grow if your SaaS solution doesn't offer room for growth. To make your SaaS scalable, you have to choose the right software stack for the project.

To help you narrow down your list, let's discuss what makes each programming language unique, including their respective pros and cons.

A Comparison of The Most Popular Software Programming Languages

Chapter 3

PHP, or if looked at long enough, short for, "Please Help"

PHP (Hypertext Preprocessor) is a scripting language used to build super simple, dynamic web pages quickly and efficiently. Since its debut in 1995 it's been a popular go-to for web developers.

Say you're working on a SaaS project and struggling to pick the right framework for developing, PHP is your best choice.

For starters, it offers the best solutions to the most common problems. Thanks to its robust app setup, each and every phase of development are kept streamlined. There are tons of packages, libraries, and frameworks that you can snag for free. Who doesn't like free stuff!?

Sit Back and Relax

You can put your mind at ease when it comes to security too since PHP is continually peer-reviewed. With the right PHP framework, you can read, maintain, and test your code a whole lot easier.

The PHP community is also large and diverse, not to mention friendly, helpful and welcoming. So if you find yourself stuck in a specific phase of your SaaS development, you can always rely on the PHP community for help.

One major drawback of using PHP is that it tends to execute more slowly compared to other programming languages. The reason for this is because the PHP framework requires more resources than most languages. Also, PHP lacks the focus of other languages when it comes to staying ahead of the curve due to its legacy baggage. Regardless of how old the language is it's still reliable and an easy go-to.

No Node.js, No Netflix, No Chill!

Node.js is an open-source run-time environment built on Chrome's V8 JavaScript engine that executes JavaScript on the server side. Huh!? What does that mean? Well basically, a long list of big companies including Walmart, Medium, LinkedIn, and Netflix has been using Node.js in running their applications. Now that's what you call an All-Star lineup!

Those who are familiar with Javascript will find it a breeze to adapt to the language used in Node.js. Since Node.js became widely used, more and more developers are able to code both on the back-end and the front-end using JavaScript. The result, of course, is getting software out the door a whole lot faster.

What To Do With All This Freedom?

On top of that, Node.js's unrestricted and dynamic nature gives you more freedom when building apps. There's so much freedom, in fact, that developers are finding building apps from scratch is totally doable.

Node.js uses a non-blocking IO system, allowing it to process several requests at the same time. As a result, your app requires less RAM, allowing for greater speed and reliability. Non-blocking IO system = less processes = less RAM used = more speed.

Node.js's main weakness is that it doesn't support multi-threaded programming so it can handle complex apps. It uses single-threaded programming, which isn't exactly ideal for processing heavy computations. Another con of using Node.js inconsistency with the API, which changes frequently. This leads to compatibility issues from time to time. Grrrr!

React.js for the likes and comments!

React.js burst into the mainstream consciousness of the programming community back in 2011 when Jordan Walke, a software developer at Facebook, used the Javascript library to power the "Likes" and commenting features on the popular social media site. So, let's all like and share React.js!

Reduce, Reuse, React.js

React.js is similar to Node.js in that it's also simple to learn if you're familiar with Javascript. One of its finest features is its ability to reuse system components, making it easy for you to develop and maintain apps. It has a virtual DOM that can streamline data updates and activities involving user interaction. With a ton of developer tools at your disposal, you get a lot of wiggle room in terms of design and debugging. Last but not least, the React ecosystem is brimming with tools that can help you fulfill your SaaS app's true potential. Simply put, there are many good reasons to develop your next SaaS solution in React.js.

One drawback of using React.js though, is that it's rapidly evolving, so much so that developers are compelled to relearn new ways of doing things. But no matter what, learning is life and whether you choose one or all three of these languages to work with, the world is your oyster!

HubSpot - All Systems Grow!

HubSpot is an inbound marketing tool that helps companies attract visitors and convert leads. It's not a programming language, but it comes with an API that can help developers integrate their SaaS enterprise apps with a dedicated CRM system. HubSpot CRM tracks customer interactions automatically – whether they're in an email, across social media, or on a call. Hubspot offers a whole lot more and what's even better, it's free!

MySQL - Structured Query Language, Just For You!

SaaS deals with information all the time, and as such requires a reliable and robust database that can access, use, understand, and process data. MySQL is one of the most popular databases used for building SaaS applications. Some of MySQL customers include Facebook, YouTube, LinkedIn, PayPal, GitHub, Tesla, Netflix, WeChat, Alibaba, Twitter, and eBay!

Who Said Nothing in Life is Free?

MySQL offers a lot in the way of functionality, and that goes even for the free version. It's easier to use compared to most of its competitors. With its vast knowledge base and mature software packaging, MySQL offers a scalable solution for enterprise applications. But there's one flaw MySQL has, it often struggles with some basic processes other systems are able to perform with little effort, including the creation of incremental backups and user management.

Python - Unlike the snake, there's nothing to be afraid of here!

Designed and made in 1991 by Dutch developer Guido van Rossum, Python is an open source programming language that is powerful and easy to implement.

It's a high-programming language that expresses functions in fewer lines of code than other languages.

It's easy to learn because it uses simple syntax rules -- mostly requiring you to use English keywords instead of punctuations. Its code is highly readable, making it easy for developers to maintain and update their web application or software.

Powerful Python Platform...Now Say That 3x Fast!

Python is powerful thanks to its robust Python Standard Library, allowing you to add a score of functionalities without writing additional code. And since Python is an interpreted language, it allows you to run the same code on multiple platforms without the need for recompilation.

One disadvantage of using interpreted language is that it's often slow. As such, it's not the ideal language when it comes to performing memory-intensive tasks. Python is also rather limited in terms of database functionality.

Ruby on Rails - Take the Train to Simple Programming

Ruby on Rails is a web development framework built for the Ruby programming language. One of RoR's most enduring characteristics is its pragmatic approach to software. It follows the DRY (Don't Repeat Yourself) principle, doing away with repetitive code and streamlining the coding process.

Stay on Track!

If you're strapped for time, RoR has rich libraries and readymade plugins and modules that will speed up the developmental process for you. It's an open source framework, so it's completely free. Another thing that makes RoR stand out is its *Rails Migrations* feature, allowing developers to make adjustments and alterations with the database on the fly.

A major drawback of RoR is that its runtime speed is rather slow. Many developers have been complaining about the boot up speed as well, although the introduction of Spring has fixed this issue to some degree. Another disadvantage RoR has is that it uses multithreading, causing requests to get queued and bogged up from time to time.

If you've narrowed down your list but still can't decide on which programming language to use, you can refine your list further by taking a look at the SaaS apps your direct competitors are using. You don't have to copy what they're doing. But by using them as points of comparison, you'll be in a better position to come up with ideas that can give your SaaS enterprise application a leg up on the competition.

Building the Right Team

Chapter 4

So, you're all finished with the MVP process. You have the basics down as far as your SaaS software's functionality is concerned. You've settled on the programming languages that will be used in developing your product.

But now you've got another challenge ahead of you, and it may be the most crucial one yet: building a team of developers to design, code, implement, and deploy your SaaS solution on the public cloud.

If you think that hiring the right person is difficult, just wait until you're tasked to

hire the right person *for* a software development team. On top of making sure that each person you hire is an expert in their respective field, you also have to check if their personality fits the company culture.

A competitive software development team needs to have at least two senior developers or CTOs (chief technology officer). When conflicts or bottlenecks arise, it pays to have one or two specialists who have the experience and the technical know-how to set things right. You need a QA engineer who can ensure that the quality of the software meets the specifications and requirements documented during the pre-planning stages. As for mid-level and junior developers, adding four to six is more or less the standard rule, but that will depend on the scope and size of the project.

Hiring In-House vs Outsourcing

There are two ways to go about hiring people for your SaaS development team: you can either hire in-house developers or you can hire the services of a specialized outsourcing agency.

To go in-house or to go with an agency? The two are entirely different beasts, with their own set of advantages and disadvantages. Ultimately, the right choice will depend on the following:

- *Budget*
- *The timeframe for the project's completion*
- *Scope and complexity of the project*

Pros of working with a software development agency

Simpler hiring process

The hiring process takes time and not to mention can be expensive. When you hire the services of a software development agency, you can do away with having to post job ads, conducting interviews, and providing training. An agency, after all, already has a software engineering team or two that can handle all of the legwork for you. Also, most of these teams have already worked together in the past, so there's a fair chance that they can work well together and hit the ground running.

Easily scalable

Outsourced development teams are easier to scale because they have staff levels that allow multiple programmers or developers to take on tasks quickly one after the other. This level of scalability allows the project to move forward even in the face of growing business demands.

Short term commitment

One great thing about hiring agencies is that it allows you to set flexible time frames for short-term goals. Once a software development team deployed by an agency is done with a particular phase of the project, their services are no longer necessary -- at least for the time being. With an in-house team, you're still required to retain your staff even if their skills are no longer necessary or applicable to your current business needs. You've already spent time and resources in hiring them so it wouldn't be practical to just let them go.

Reduced costs

The most obvious advantage of hiring outsourced teams is the reduced costs involved, especially when you're hiring offshore. If you hire the services of a BPO company from Armenia or the Philippines, for instance, the labor costs involved will be less than half the amount you would otherwise spend when hiring in-house employees from the United States. In addition, overhead costs are reduced to a notable extent since you're not required to provide the equipment and the office space necessary to run the operation.

Cons of working with an outsourced software agency

Quality control

As the project manager, you have to oversee the project at times to ensure that everything's going smoothly and according to plan. You have to make sure that every cog in the machine is primed to deliver on your project goals. Additionally, there will be times when you need to discuss attendant challenges with your software engineers. When you hire an outsourcing agency, you don't have that amount of control, and it's easy for a project to lose its direction without proper guidance.

Collaboration

One significant disadvantage of hiring an outsourced agency is that the team of developers tasked to work on the project may be using English as a second language. When there's a language barrier between two parties, important information could get lost in the translation, which could negatively impact the quality of the software. Make sure that you hire an agency with competitive speaking management skills to ensure that communication is always crystal clear and precise.

Logistical challenges

Differences in time zones can pose logistical challenges especially when the project requires a quick turnaround time and close collaboration. Obviously, scheduling a call is difficult when your team is on the other side of the world. One good way to overcome this issue is to set at least 2-3 hours of time zone working hour overlap.

Commitment to the project

Outsourcing agencies work with multiple clients, not just you. This, of course, means that the team working on your SaaS app might also be working on other projects. As a result, they may not be as committed to the project as you are.

Security and confidentiality

Protecting sensitive information or data is hard enough when working with an in-house team, and even more so when you're collaborating with an external agency located in a different time zone. On the other hand, working with an in-house team ensures that your company's information is safe and secure because of stricter security laws and penalties in your own country.

Pros of working with in-house software development

Long-term product development

Close-collaboration over a long period of time can have a positive cumulative effect on any project. There's this clear sense of urgency that you get when you're working with a team under the same roof. This instills long-term commitment and allows your team to know the project inside and out.

Aligned interests

An in-house employee knows that he or she represents the company and as a result is more likely to share the organization's mission and objectives. When an employee is able to identify with the company, they will be more motivated to contribute their skills for the betterment of the organization. The fact that their interests align with the company motivates them to work harder, working on the project as if their careers depended on it.

Company's culture fit

A startup company with employees who share the same values and vision is more likely to be successful in implementing even the most difficult projects. A good business owner always makes sure that the company's culture stays consistent and is shared. Such a thing is hard to do when the team of developers you're working with belongs to a culture far removed from yours.

Shorter turnaround time

Developing software in-house enables you to closely monitor each phase of the development process, allowing you to spot issues as they arise. As a result, these

problems can be addressed instantly, allowing the project to be completed at the soonest time possible.

Alignment with the company‘s standards

Collaborating with developers in-house ensures that your startup’s assets are well-maintained and closely-monitored. With an in-house software development team, you can decide what’s best for the future maintenance processes of the product.

Cons of working with in-house software development

Time-consuming process

Managing in-house developers comes with a wide range of responsibilities. For one thing, the recruitment process itself can take months, and the costs involved with interviewing candidates can take its toll on your financial resources.

Technical expertise

Specialized firms have the infrastructure, processes, and budget needed to recruit experts with deeper experience. Unless your company has a large IT department or an unlimited budget needed to hire a wide range of IT professionals, chances are you won’t be able to cover all your specific needs with dedicated experts.

Limited funds to hire a talent

Hiring a specialist on a full-time basis can take its toll on company resources. As is the case with most startups, there will be times when business requirements don’t call for the services of a specialist. By continually paying for services you’re not able to maximize, your bottom line will almost always take a major hit.

Side costs

Hiring in-house employees, as mandated by law, requires you to pay for side costs, including annual salaries, benefit packages, overhead costs, taxes etc.

Hiring outsourced programmers, on the other hand, takes this responsibility out of your hands.

Overload of in-house IT resources

Managing in-house employees and resources over a long period can potentially overload your IT department's resources. More often than not, in-house employees have no choice but to keep up with a growing number of projects with limited resources. This places a burden not just on your budget but your employees as well. This overload usually results in unsatisfied employees and, in turn, a lackluster product.

Managing a SaaS Development Project — Incredo's Process

Chapter 5

In just three years, Incredo has become a global leader in the SaaS marketing industry. Thanks to our dedicated and talented team of managers, designers, marketers, and developers, we continue to deliver reliable and innovative solutions to a long list of satisfied clients around the world.

Our success, however, did not come without its attendant challenges and growing pains. We've learned from our initial failures and treated them as opportunities

to refine our strategies and processes? The result, of course, is steady and rapid growth over the years. We at Incredo have mastered the art of managing

moderate to highly-complex developmental projects, and a big part of that can be attributed to our Agile approach towards management and digital transformation. Out of our deep sense of gratitude to our clients, we'd be more than happy to share our expertise and knowledge on how to manage a SaaS development project towards sustainable success.

Managing a SaaS development project needs more than just keen attention to detail, it also requires an exceptional management acumen, a strong commitment to consistency, a collaborative approach, and sometimes, dogged determination and grit.

To guarantee that every phase of the development project will go according to plan, you have to ensure that each member, from the top down, is aligned with the project's objectives and goals. Each outcome should contribute to the big picture, with specific timeframes set up for well-defined actions and deliverables.

Planning the execution

To get a clear picture on how to manage a successful SaaS development project, we're going to break down the steps and the facets needed to bring the overall plan to fruition.

Identifying the project

A SaaS development project has many moving parts. If you don't define the business goals from the get-go, there's no telling where those moving parts will take you. So, you need to identify the project from the start. You need to take all the information from the user requirements and make sure that the goals you've set for the project are aligned with the needs of your target users.

Defining goals and objectives

Every great SaaS project, or any project for that matter, starts with an idea and a vision of what's possible. To turn that idea or vision into reality, you need to have a clear destination. You have to find your North Star and then follow it, so to speak. But if you want your team to be on the right track, you should communicate your vision in a simple, clear, and effective way. You can do that by defining your SaaS developmental project's goals and objectives.

To accomplish this, you need to ask the following questions:

- *Why are we working on this project?*
- *What problems are we trying to solve?*
- *What criteria will be used to judge the project's success?*
- *Who has a stake in the outcome?*

By answering these questions, you can come up with specific goals that are worth pursuing as well as clear objectives that can serve as a roadmap towards achieving them.

Task breakdown

A detailed breakdown of all tasks ensures that every single one of them has purpose and is aimed towards executing and achieving your project's goals. Adopting the SMART principle is a great way to give your project the clarity, focus, and the momentum it needs to get to the finish line. To achieve S.M.A.R.T. goals, you need tasks that qualify for the following criteria: *specific*, *measurable*, *attainable*, *realistic,* and *timely*.

Building the team

A SaaS app is only as good as the development team who builds it. As the business owner, it's in your best interests to build a team that can think and act like a sum of its members. You need to make sure that each member is an expert in their field and has the right personality for the project.

Identify potential project killers

A long-term project can only be successful if the project leader always takes the time to prepare for the worst. By anticipating and identifying potential issues the project may face, you and your team will be more prepared to solve any crisis before it's too late.

Timeline creation

To keep the project running smoothly, all tasks need to be broken down into stages so that they can be completed within the allotted time frame. To set realistic deadlines, the project manager should discuss the scope of the project with the team and be transparent. This way, expectations can be properly set.

Get feedback

After all the planning is done, ask for feedback from teammates, stakeholders, and clients. The goal here is to make sure that everyone's on the same page. Better to lay out all the cards on the table and get everyone's approval rather than move ahead and then back peddle to make adjustments again at a later stage.

Adjust the plan accordingly

Even the most efficient of project plans won't account for unforeseen problems that may arise. Have some flexibility with your plan so that you can adjust and plan accordingly. A good way to do this is to examine external influences that may affect your project going forward and then map out a contingency plan for each of them. Make sure to communicate your contingency plans to your team so that they can spring into action as soon as problems arise.

Keeping the project on track

As established, managing a development project for a SaaS app comes with a fair share of challenges. It's a complex undertaking that requires close collaboration, efficient planning, a strategic mindset, and a strict protocol.

Here are the best practices that will ensure that the project is moving ahead as intended.

Regular interval planning for checking progress

Your list of plans won't do much good if you don't revisit it every now and then. Once you've mapped out every phase of the plan, you have to look at it every week (or a few days, depending on the interval you've set out for each plan) and determine what your goals are for each specific week. Examine the steps you and your team need to make to bring that phase of the project to completion. Better yet, you can discuss the plan with your team at regular intervals to keep those goals in mind. Making sure that your project's objectives are top of mind ensures that each job is done with a sense of urgency.

Be open to suggestions

A project manager should have the humility to accept that he doesn't know everything. You may have a vision and a goal in place, but in the final analysis, your SaaS development team has the expertise and the experience needed to make it a reality. If your developers have suggestions, you'd be doing the project a favor by paying attention. Giving their suggestions the consideration they deserve could potentially advance your SaaS development project in ways you haven't even considered before. We all have two ears and one mouth for a reason. Be a good listener.

Keep an eye out for possible obstacles

Every plan needs a backup plan. A plan B, C, or even more. Once you're in the developmental stage, you and your team may face obstacles that hadn't been accounted for during initial planning. Make sure that your team is primed to take action for every possible eventuality.

Lead the team

As a leader, your role is crucial to the successful implementation of your SaaS project. Guided by the company's vision, you need to oversee the operation from all facets and guide your team accordingly. You have to provide your team with the appropriate tools and resources to help them do their jobs to the best of their abilities.

Using project management software

A project has everything to gain by using technology to its advantage. Thankfully, there's no shortage of project management tools that can help you optimize your project tasks, from the planning stages to task implementation to the final product release.

These tools are instrumental in bringing about the proper execution of tried-and-tested methodologies. If you work in the tech industry, chances are you've heard of the terms "Agile" and "Scrum." The agile management system is often spoken of in reverential tones. And it's just as well. By assigning a "scrum" to guide your team, every stage of the project development will be executed in the most efficient and effective manner.

Below are the project management software tools we recommend.

Teamwork Projects

You need to make sure that all hands are on deck for the project to move in the right direction On that end, you can use Teamwork Projects to ensure that all members can easily monitor their tasks. As a business workflow tool, Teamwork Projects is peerless, allowing you to communicate and coordinate seamlessly with your teams. It's got a wide range of collaboration tools to keep your team engaged and productive throughout the entire project. While its search capabilities are rather limited, Teamwork Projects is a nice tool to have for projects that require a team-oriented approach.

Trello

Trello is another reliable management app that helps you monitor a project's progress. By giving you a visual overview of the tasks involved as well as the people working on them, the software helps you make informed decisions and take appropriate actions to further your project's development. The visual configuration is intuitive and functional, making it easy to set up and manage projects, facilitate team collaboration, and manage to-do lists. Loading can be a pain at times, and creating new cards from a spreadsheet could be less sluggish. But in the final analysis, Trello is a great project management tool that is worthy of consideration.

Asana

Asana is an app that takes the guesswork out of project management. Using the Timeline feature, you can use the app to monitor projects and tasks through every stage, as well as help you coordinate the activities of your team. It also comes with an excellent "to do" list, allowing you to stay on track of your team's progress in every phase of the project. Asana's "List" reports can be confusing at first, but it becomes less so the more frequently you use it.

Basecamp

Basecamp prides itself on its ability to put everything you need to get work done in one place. Such a value proposition is hard to ignore for any project manager who wants to bring about project completion in the most efficient manner. For one thing, it comes with a task viewer that gives you a bird's eye view of the entire

project. Navigation is intuitive, so much so that you don't have to search through every folder to find what you're looking for. Team communication is functional and efficient thanks to its dedicated notification board. One drawback Basecamp has is that it's rather limited in features, especially in terms of customizations and reports.

Atlassian

Atlassian's Jira software is the perfect tool to carry out Agile initiatives within your team, allowing you to plan, track, and implement a project efficiently and with minimal risk to project resources. It's got a wide range of features to make the lives of project managers so much easier, including advanced search query, view-in issue navigator, follow issue, linked issues, and more. Granted, the clunky interface will take some getting used to. The inefficient workflows could also use some work. But overall, it's a great software to have if you want to complete your projects with less wasted time and effort.

Conclusion

The SaaS market is ripe with opportunities, and you'd do well to take advantage by coming out with a fresh and unique solution that will put your brand on the map. It's important to remember that the process of SaaS development must be backed by a sound policy framework and a definitive objective so that your team can make strategic decisions. Otherwise, you'd be drifting from one milestone to another without accomplishing much. Having skilled developers and programmers who are fully involved in the entire process will also be a tremendous help.