

Guide for

SaaS Application Development



Table of contents

SaaS Stats	03
What you should consider when developing a SaaS application.....	04
How to develop a SaaS application (step-by-step).....	07
SaaS development team composition.....	11
Cost of building a SaaS application.....	15
Your syoft partner.....	12
FAQs.....	14

Perhaps the logical next step on your path to business success is creating a SaaS application. However, you shouldn't take this decision carelessly or without doing your homework first.

To hire a development team, allocate funds, or advertise a new product effectively, you must have a thorough understanding of the SaaS development process. And this is the topic of this article.

By creating SaaS solutions for growing and established organizations, we at Syoft have contributed to the success of more than 20 firms. So you can rest easy knowing that the advice in this article has been tried and true.

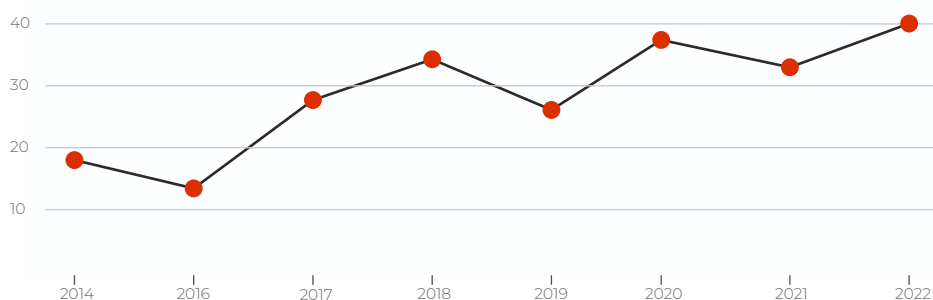
Learn more about SaaS application development best practises and other information about outsourcing and SaaS application development by reading on.

SaaS Stats

The cloud shift rate through 2020 for SaaS is 37%—that's three times more than PaaS (Platform-as-a-Service) and IaaS (Infrastructure-

as-a-Service). And according to BetterCloud, more than 75% of organizations claim to make 80% of their business apps SaaS in the next five years.

SaaS will account for 80% of all business applications by 2020



Basic set of SaaS app features

- Cloud-based SaaS solutions must have a specific set of features that set them apart from other kinds of online services and platforms if they are to provide a comprehensive experience. These consist of
The ability to login and logout of your user profile
- Flexible charging structure based on subscriptions
- Application and data security
- User-friendly interface
- Automatic updates
- Option to get email notifications

What you should consider when developing a SaaS application

When developing a SaaS application, there are many different factors that must be considered. These factors can be classified into two groups based on their level of technical complexity.

Technical characteristics of SaaS applications

A solid SaaS application should perform flawlessly in the following aspects:

Databases

Having a database that is both sizable, adaptable, and secure is crucial, but it must also be tailored to the specific requirements of the business. The selection of the appropriate database type is influenced by a variety of factors, such as the data format, scalability, expected traffic, the ratio of read-to-write operations, programming language used, and available financial resources.

Scaling

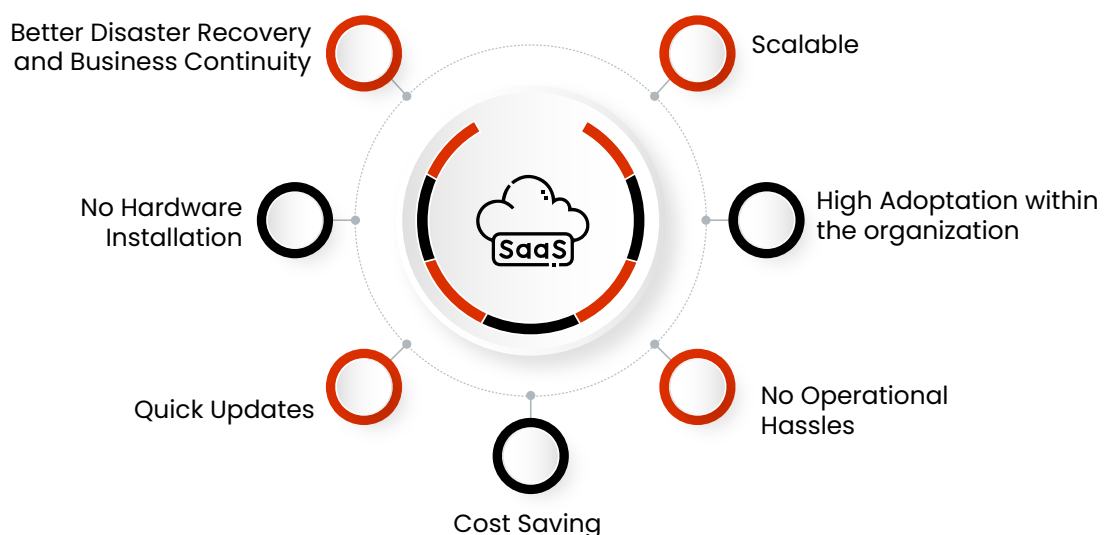
From the outset of SaaS development, it is important to consider scalability as an integral aspect. The software being created must have the ability to support both horizontal and vertical scaling, with business growth being easily accommodated without any issues.

Frequent updates

Software cannot be perfect, as clients' expectations and requirements are likely to increase over time. In order to meet these evolving needs, regular updates to the SaaS solution based on user feedback are necessary to maintain relevance.

Third-party integrations

Enabling integration with third-party software enhances the appeal and functionality of your solution, providing added value to users. To facilitate such integrations, it is essential to create straightforward APIs and accompanying documentation within your software. Integrating popular platforms like Slack into a SaaS application can be highly advantageous.



Non-technical characteristics of SaaS applications

A SaaS's success is not solely dependent on its robustness, as the other half of success comes from user-friendliness. Therefore, ensure that your solution incorporates the following features:

Non-technical characteristics of SaaS applications

A SaaS's success is not solely dependent on its robustness, as the other half of success comes from user-friendliness. Therefore, ensure that your solution incorporates the following features:

Market fit

Before beginning the development process, it is essential to ensure that there is an actual market demand for your solution. Conducting market analysis and determining your solution's position within the market early on will enable you to establish realistic expectations and proceed accordingly.

Client acquisition process

When building an app for users, it is important to prioritize ease of use and visual appeal, as well as ensuring that it is marketed to the appropriate audience. Demonstrating concern for clients also involves developing a subscription lifecycle and establishing a fair pricing policy.

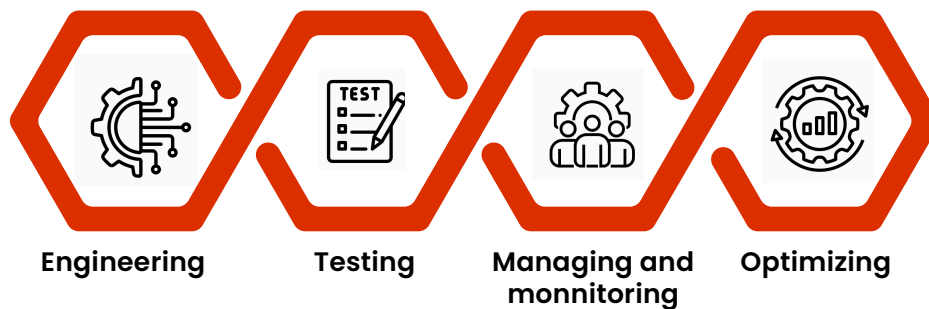
Various packages

Providing various subscription options increases the flexibility of your solution and can attract a wider range of clients. Additionally, users will appreciate the ability to continue using your SaaS application even if their budget changes, by simply upgrading or downgrading their subscription package.

Now that you are aware of the necessary features and technical/non-technical considerations for your SaaS application, you can finally begin coding.

How to develop a SaaS application [step-by-step]

The process of building SaaS applications usually follows the following steps:



Engineering

The most substantial portion of work in SaaS software development is building the application from scratch. This process typically requires several months and a team of dedicated experts, each focused on their respective field of expertise.

Design

Simply beginning the construction of cloud applications without first designing the solution can be costly, as it risks producing unstable software that includes unnecessary or faulty features. The design phase is critical and involves multiple cycles, resulting in essential artifacts such as the software architecture document, user stories, style guides, mockups, and, occasionally, a functional prototype.

The key aspects of SaaS solutions can be focused on, starting with the hosting architecture of the application and its user accommodation. The option between a single- or multi-tenant approach can be chosen.

In the single-tenant architecture, a dedicated server is assigned to each client when they use the software, making it more suitable for larger clients who may rapidly consume resources.

On the other hand, the multi-tenant architecture allows multiple users to access the same database through separate accounts, ensuring that they remain unaware of each other's existence. There are two methods for implementing the multi-tenant approach:

One app instance, one database. With this setup, all users entering your cloud environment access the same database until it's full. While this approach is easy to implement, its scaling abilities are limited, which affects the overall performance of the app and the user experience.

One app instance, several databases. Here, each database is only filled up to a certain point before redirecting new users to another database. That way, users have access to more resources, so the software feels more responsive. This approach is much more expensive to implement and requires more resources early on.

The next step is choosing your software development tech stack. When selecting the tech stack for software development, it is important to consider various factors, such as flexibility, scalability, budget, and speed, as this is the foundation of your web app.

Partition

Separating the data into distinct partitions or data stores is crucial to accommodate future software modifications, including scaling, performance improvements, operational flexibility, and enhanced security. By breaking down the data into partitions, it becomes easier to handle each partition separately, reducing the complexity of the entire database.

There are various partitioning strategies available, such as horizontal, vertical, and functional partitioning, each with its advantages and disadvantages.

Deploy

The deployment of software can be compared to opening a store's front door. It requires everything to be set up properly, including easy access to comprehensive documentation and immediate support options for users. To maintain this state, automation like a factory production line is necessary for deployment. Software updates are rolled out in real-time, getting released as soon as they're ready. This automated deployment approach is different from routine updates in terms of speed and flexibility. Instead of releasing one major update every so often and putting your entire system on hold, you can deploy small changes incrementally.

- Deployment automation is the engine of SaaS agility
- Code and data must evolve in real-time
- View this as your competitive advantage
- Expect an ongoing investment in tooling and automation



Testing

In SaaS software testing, the goal is to ensure that the software meets both your and user requirements and has minimal bugs before and after release. To achieve comprehensive coverage, it is recommended to use a combination of manual and automated testing approaches in your quality assurance process. Additionally, beta testers can help uncover unique use cases that you may not have considered. To bolster security measures, cloud security assessment can also be incorporated into the testing process.

Managing & monitoring

The development process operates more efficiently when a manager oversees the team's coding and testing activities. The manager is responsible for assigning tasks to team members based on their skill level and guiding them through the process. They provide regular updates on the SaaS development progress and ensure that the team is on track to meet their goals.

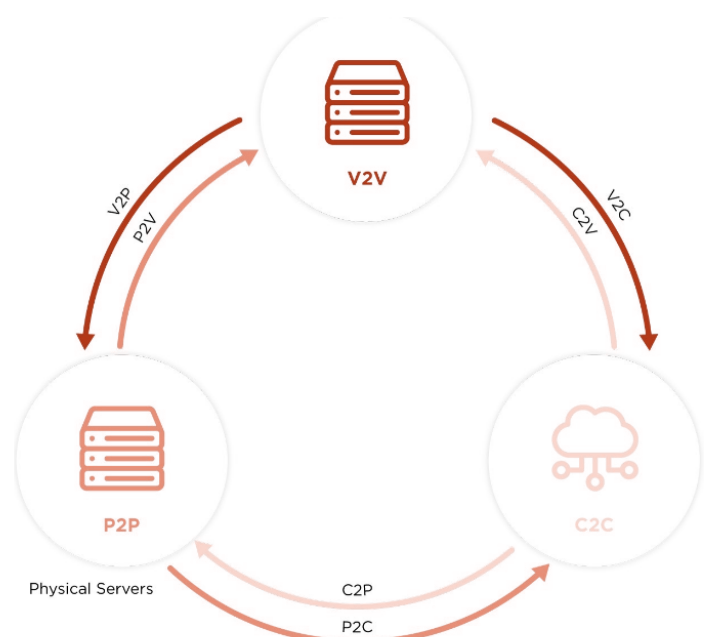
Optimizing

Optimizing SaaS environments involves the optimization of costs, tenant experience, availability and performance, timing, and bulk operations. It is crucial to perform this optimization after scaling to the desired size. To ensure a smooth-running software, it is essential to identify the necessary resources by ensuring sufficient servers and databases for the user base. Balancing resource allocation is key to ensuring a pleasant user experience and avoiding unnecessary expenses.

Migrating from on-premise to cloud

Unless your SaaS development framework was cloud-based from the start, it needs to be moved to the cloud. Depending on your server type, amount of data, and acceptable downtime, there are multiple migration options to choose from:

- P2V (physical-to-virtual)
- P2C (physical-to-cloud)
- V2V (virtual-to-virtual)
- V2C (virtual-to-cloud)



SaaS development team composition

Developing cloud-based SaaS software requires specific roles to be fulfilled. The required roles include a project manager, business analyst, UX/UI designer, backend developer, frontend developer, and QA engineer.

- The **project manager** is responsible for planning the development process and ensuring it adheres to a set timeframe and requirements.
- The **business analyst** analyzes the software's role in the market, documents its progress, and builds a viable business model.
- The **UX/UI designer** designs and implements the application's user interface and experience, ensuring it's easy-to-use, aesthetically pleasing, and aligns with the brand.
- The **backend developer** works on the server side of the application, ensuring it functions as planned using scripting languages.
- The **frontend developer** works on the client side, turning designs into code using JavaScript and its frameworks.
- The **QA engineer** tests the software for defects, bugs, and issues and reports them to be fixed by the engineers.

Depending on the project's size and budget, other specialists like **DevOps engineers, technical writers, marketing specialists, or product managers** may be needed.

Partnering with a software development vendor can offer access to a wide variety of specialists and expertise.

Cost of building a SaaS application

Most software development costs are attributed to the salaries of software engineers and other essential team members. When building an in-house team, it's important to consider the costs of hiring and onboarding new tech specialists when creating a budget. On the other hand, outsourcing software development includes paying for the hourly rates of engineers, in addition to the vendor's fee.

According to YouTeam's recent survey, the average hourly rates of senior software developers around the globe vary drastically:

- United States: \$46.32/hour
- United Kingdom: \$71/hour
- India: \$30/hour
- Ukraine: \$37/hour

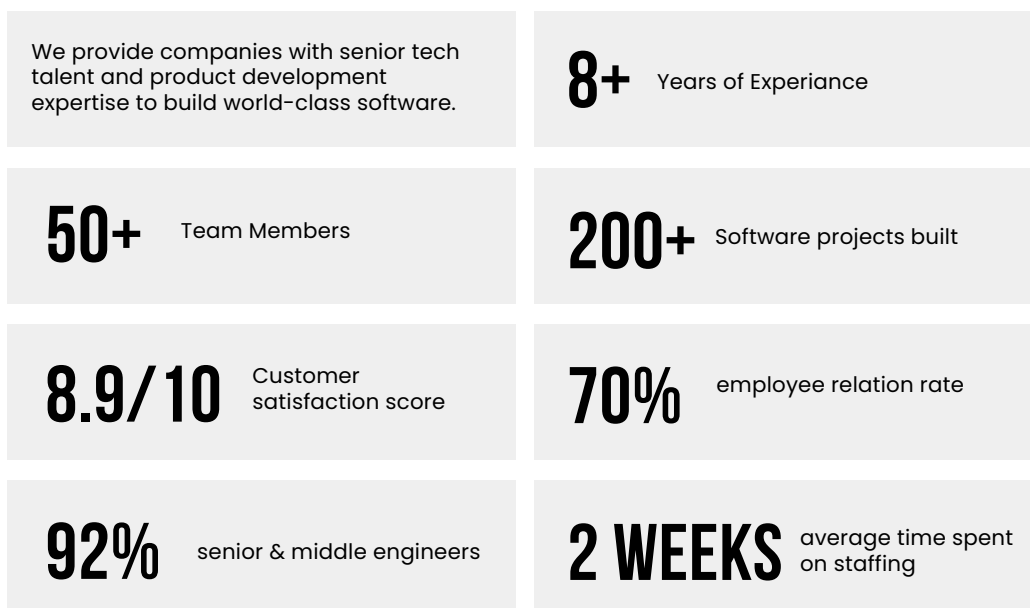
Speaking of Ukraine, outsourcing SaaS application development to a vendor from this country might well be the optimal choice in

terms of the quality-cost ratio. Ukrainians share the Western business values, are educated and hard-working, and live in a time zone that's reasonable for clients from both the EU and the US.

Popular SaaS apps



Your Syoft Partner



If you decide to delegate your project to a SaaS development company, it's important to ensure they have the necessary expertise and experience and can apply that experience to your project. Luckily, Syoft is a trustworthy outsourcing vendor with seven years of experience in SaaS development. Syoft has worked with high-growth SaaS companies from around the world, including:

Worke: a product that helps small & medium business to accelerate their business with super conversational chat bots & CRM

Zepul: An AI based Applicant tracking system with a market place for different stake holders in the recruitment space.

Eazyrooms: a SaaS application that helps increase operational efficiencies of hotel properties and elevate Guest Experience with super-conversational chat bots.

Tazku: A task and client project management system for service-based businesses with automated reminders and other business automation applications.

Whenever you decide to add your name to the list, contact our experts. They will be happy to help you build a SaaS application your users will love, and you'll be proud of.

How to choose a SaaS development vendor?

When selecting a vendor for SaaS application development, it's crucial to consider their experience and expertise. Here are some steps to take:

- Check the vendor's portfolio and case studies to see if their previous work is similar to what you have in mind.
- Look for client testimonials and reviews on platforms like Clutch and Goodfirms to get an idea of what it's like to work with them.
- Evaluate the vendor's communication and project management skills to ensure they can effectively collaborate with your team.
- Assess the vendor's technical capabilities, including their proficiency in programming languages, frameworks, and tools relevant to your project.
- Consider the vendor's location and time zone, as well as their cultural and linguistic compatibility with your team.

Why is multi-tenant architecture the optimal option for SaaS application development?

Both forms of multi-tenant app architecture (separate database & shared database) are less expensive, more flexible to expand, and better at managing and accommodating additional tenants than single-tenant architecture. It enables you to begin with a small scale and grow as needed while still delivering a satisfactory software experience to your customers.

How long does it usually take to develop a cloud- based SaaS application?

The duration from the initial planning phase to the launch of the product may range from 2 to 10 months depending on various factors, including the complexity of the application, its scale, the technology stack used, and the size of the development team.

How much does it cost to develop a SaaS application?

The expenses associated with developing a SaaS application are influenced by the location of the development team. For instance, recruiting a team from the United States might incur an expense of approximately \$350,000 per app, whereas shifting the development process to India could decrease the costs to around \$120,000 per app.

Interested in developing a SaaS application?

**Contact us to discuss the cost
and ways of cooperation**

Email us to **welcome@Syoft.software**
Or fill the form at **www.Syoft.software**