The Guide to Software Product Development

In this guide, ScienceSoft highlights everything you should consider before embarking on your software product development project, from theoretical particularities to hands-on organizational tips.

Key Aspects of Software Product Development

UX is critical

User personas. List all possible users of your software and think about their use purposes. By imagining how exactly each persona will interact with the product, you create user scenarios that will help you see what features could drive people to buy your product.

Intuitive UI. No matter how complex the functionality behind any scenario may be, the product should not impose this complexity on your user. Make sure your UI isn't overwhelming and the navigation is self-explanatory. The visual appeal is important too, for both catching the attention of your users at the start and investing in their long-lasting positive impression of your product.

Example: if you're creating software for a medical device, you'll need to think of its possible users (patients, doctors, nurses) and their different intentions when using a device. Take the age of your users into account too: if your user base includes children and the elderly, you may want to make your features more easy-to-understand or even add a simplified UI version.

Demand-based feature choice

If you try to satisfy all the various user needs, your product will be too complex and won't be successful on the market. It's important to limit your functionality, so a good option may be to create an **MVP**. Take a number of features that are most common among the scenarios for different user personas and make them the core of your product's initial version.

Quality and security

Even if you fix the errors in an already released product with a downloadable update or by offering the next version, your users will most likely have formed a negative image of your product and won't trust enough to invest in your product.

To deliver quality and secure code, build your product development process around the principles of a **Secure Software Development Lifecycle policy**, such as Microsoft's SDL.

Constant evolution and delivery

The evolution of your product allows you to gradually cover more user scenarios. Consider using **application performance monitoring** to get information on the users' needs and expectations for the next product versions.

Technological Aspects to Consider

There're also a number of tech strategies that don't necessarily suit every project but can be considered individually.

Software-as-a-Service (SaaS)

The SaaS model implies providing online access to the software product hosted on cloud servers, which saves you a lot of product distribution troubles. Moreover, the model doesn't narrow your target audience to users of specific hardware or platforms, as all a customer needs to use your software is the internet connection and a subscription for your product. Your advantage is also the possibility to set flexible pricing options and deliver updates instantly.

Application monitoring

Application monitoring is a strategy that lets you track the performance of your product in use and get data on customer behavior. For instance, you can retrace code transactions in order to understand and address a user's performance problems. Moreover, you can get reports on how your users interact with your software to better understand what they expect from your product's UI and functionality.

Integration with specific hardware

If you need to develop software for your company's hardware (IoT smart appliances, industrial machinery, medical equipment), you need to keep in mind that a development team will most likely need access to your hardware product for successful integration.

If your hardware is expensive to ship, a local vendor is probably a better fit. At the same time, the vendor's experience with your type of hardware can compensate the shipping costs.

Mobile availability

If your product isn't already aimed at mobile devices, the market will expect it to offer mobile availability. There're several ways to introduce it:

- Adapt it for a mobile screen (in case your product is a web solution).
- Create a native or cross-platform mobile app version of your software (you may want to include only key features to keep the app lightweight).
- Develop a complementing app to control your software product remotely (in case your software is intended for specific hardware).

Organizing Product Development from a Business Perspective

Whether you choose to develop your product with an in-house or an outsourced team, you'll have to organize the development process from the business perspective. Let's take a look at the major points that are important to take into account.

Managing costs and time to market

Design-to-Value. Based on your individual vision of 'value', you can choose to make user interest, competitiveness, or brand image your ultimate goal that will define the feature choice of your product. Thanks to this laser-focus approach, you'll be able to discard secondary features and cut the time and costs of product development.

Reuse of components. Creating everything from scratch isn't always necessary, and realizing that your product can use an already available framework, platform or services is crucial before the development launch. Make sure you reuse as many components as possible to cut your costs and development time.

Versioning

Even when preparing your product for its very first release, you have to think one or two versions ahead. Of course, each of the iterations will have certain corrections based on your users' feedback, but by making past mistakes always drive your versioning, you won't let your product truly evolve. Let the feedback help you adjust what was done wrong, but don't limit yourself to making fixes only.

Managing risks

Take different risk areas into account – time, budget, performance, etc. – and carefully estimate each of them. After prioritizing the risks, basing on their probability rate and potential damage, you'll be ready to develop an individual strategy to address each of them.

The key of risk management is to keep ownership of your product – that is, to have rights and access to the initial code, tests, configuration files and all the necessary documentation. This way, you will always be able to end your partnership with a problematic vendor and continue from where you left off with a different software product development company.

Takeaways

✔ Before you choose the functionality pack, make sure you know your target audience and have a list of future user scenarios.

✔ Focus on quality and security to avoid after-release hurdles.

✔ Depending on the type of your product, consider SaaS and hardware integration, as well as give application performance monitoring and mobile availability a serious thought.

✔ Don't lose sight of the business aspects. Save your costs by concentrating on the benefits from your product and casting aside everything that doesn't get you closer to them.

✔ Keep time, budget, performance, and vendor risks in check and be prepared to answer them.

✔ Start thinking about the future versions of your product from the very start and always stick to the evolution plan while fixing past errors and defects.

Looking for a vendor to develop your product or join your project as a tech consulting partner? Get in touch with us at contact@scnsoft.com for a free consultation of your case.