## Introduction

"If you can't waste time on SaaS, you can't make money on SaaS." It's a cliché. As a software industry practitioner, I often encounter this problem. For our SaaS products, we all know that there is a "lifecycle" concept, that is, every month a new product (software) has to go online, so what you have to do is to try to make the features as good as possible, so that the user experience is more smooth and perfect, so that you can ensure that there is not too much loss. But how is this "life cycle" calculated? For general enterprises, we do not have the ability to calculate the time and number of customer use, every month we need to provide new products (software), new features for enterprises? The answer is no. Because there is a cycle in the middle called "life cycle". That is, if we divide one month, one year, three years ...... into segments, we can calculate how it is calculated. But in fact, most of our companies do not divide the "life cycle" of products in this way.

## User experience first

The first thing we need to know is the principle of user experience first. Why do I say this? Because when designing a SaaS product, we first need to understand the market and customer needs, and secondly, we must make the user experience good enough to be recognized by customers. When we incorporate this principle into the product design, we can make the product more "easy to use". This can be achieved by making the product more "easy to use" in the following ways: the user can easily get started; the operation process is simple and clear; and the functionality meets the user's needs. In this way, you can better design SaaS products to do a good job, do a good job, but also to avoid a lot of unnecessary "waste".

## The business process is simple

If the product design is not reasonable, then it will bring too many problems to customers. For example, when we do enterprise management system, if we use the traditional development method to design the product (software), it will increase a lot of workload, and will make the user's operation of the system becomes more complex. I once saw such a few people in an enterprise who were writing a product (software), and all they did was write code, draw flowcharts, and look at their requirements again. Of course they got the job done, but they thought the product (software) should be done that way and not so complicated. And we also encounter the same problem, that many features seem to be perfect, but in reality it is very difficult to implement. If you are not a business person, then you may not know whether what you have designed is practical for customers or not.

1、The process design should be as simple as possible

If your business process is too complex, then it will become very difficult for users to use the product (software). For example, we can imagine when a restaurant wants to go to a customer for a visit, they need to walk from the entrance of the parking lot to the restaurant to sign in. It doesn't sound complicated, but it's actually very tedious. Similarly, when we have a meeting in the office, we need to sign in in the conference room, and each person needs to sign in twice. Such a process design looks very simple, but in reality it is very complicated because it not only takes up a lot of time for us to do this work, but also affects customer satisfaction with the service you provide. On the contrary, if your business process design is simple enough, then the customer will not feel satisfied with the services you provide.

2、Minimize "useless" functions

The "useless" here does not mean that these functions are not useful, but that they are not useful to customers. For example, if we can let employees see their own performance information when logging into the system, it would be good. Another example is the "time clock" function that I am using now, where users only need to enter their name, department and employee name when they clock in and out. I found that this is the same for sales, because they just enter a name; and sales don't really care how many people are in the company. If you can bring this convenience to customers, then they will definitely like your product (software); conversely, if the user does not care about these, then you need to consider these issues when you design the product (software).

## Do not affect the user

This is also a principle that I have been emphasizing when making products. The user is our bread and butter, so you can not serve the user when you make the user unsatisfactory behavior. Here many people may find it difficult to understand, "customers" in the end who is? Of course, customers do not refer to general corporate customers, large enterprises, group customers and so on. Rather, it refers to all the employees of our own company. So if you do not understand your own company's products (software), then do not waste time and energy here. Of course, if the product has too many complicated and tedious operations, don't do it either. Of course, the requirements of "user experience" are not only these, but if you don't do well in this area, it may also affect other colleagues' view on your design concept and make you rejected by others. So here's a reminder: don't make yourself look like a bully! We have to give our customers more choices and freedom of leeway. We can define the service user as a "service provider" role. So when a company puts forward some new needs, we need to understand what he actually wants to solve the problem, and then design a solution to him, or you can communicate with the customer to help each other find a solution; or directly to the user can be proposed. In addition, when designing products or features, we need to take into account the changing needs of different types of businesses. For example, like e-commerce platform to provide users with a large number of optional commodity information, then it is certainly not.

## Reduce the amount of “useless functions”

Our products do not need to develop any plug-ins, as long as you design it more simple and convenient, then customers will also feel that your products are better to use. (This is also why more and more SaaS software will appear in the "enterprise WeChat", "nail" on the reason. Because if you integrate a lot of plug-ins together, it is more convenient for users.) -For example, we have done a project, it has a lot of functional modules, but in fact each module only needs to use a few points; or some functions in our design only as a core function to design, but when we use it found that the user needs to do a lot of manual calculations; or some processes are very cumbersome, the user needs a lot of manual calculations to complete, and (These cannot be designed as core functions.)