SDK is an important part of TOB products, which can be simply understood as providing programmers to help them create software. So how do you design an SDK product and what are its design tips? This article will reveal them all.

SDKs are unfamiliar to many non-technical product managers, but for the ToB product space, many ToB products need to provide their own SDK products for company-level developers.

Of course, SDKs are first and foremost developer-oriented, so it is more accurate to call them ToD products, but in terms of the broader scope of ToB products, SDKs are part of the broader ToB product category, so they are included in the broader ToB product category here.

So what is an SDK and what is the design of an SDK? Let's break it down one by one.

I. What is SDK?

Wikipedia's definition of SDK.

SDK: Software Development Kit (SDK) is generally a collection of development tools used by software engineers to create application software for specific software packages, software frameworks, hardware platforms, operating systems and so on.

The definition of SDK is rather tongue-in-cheek, so let's simplify it: "SDK is a collection of development tools used by programmers to create software".

This is clear, let's clarify a few key points.

1. programmers

SDK is developed by programmers, but also to provide programmers to use. Can be said: from the developer to the developer.

2. create software

SDK is used to help programmers to create software, used to help programmers programming.

3. Collection

SDK is a collection of development tools, here to emphasize that it is a collection, it contains more than just functional module files, it also contains the use of these library files interface, description of the documentation, the use of the demo and other content.

In short, SDK is a kind of existence.

If you are a product planning student of a ToB company, and your company provides some functions to the developers of the client company, these functions may be QQ login component, Alipay payment component, communication channel component --- and other kinds of modules, these function modules provided by your company exist in the form of SDK, and provide to the programmers of the client company.

After the programmer of the client company gets the SDK from your company, you can use the library files (lib format, dll format, so format, etc.) in the SDK to program and integrate the functions provided by you into your own software according to the description document inside, so that their software has the functions provided by the SDK.

II. What are the common SDKs?

1. Under Windows system

there are many SDKs for developers to use to develop various kinds of products on Windows system.

2. Under Android system

There are official Android SDKs launched continuously for Android developers to develop various products on Android system.

3. SDKs provided by third-party companies

For example, in 2010, Tencent provided open strategy to open Tencent's social capabilities to the industry, that is, our common QQ sharing, WeChat friend sharing, friend circle sharing and other functions, these functions on Android system can be provided to developers in the form of SDK.

Here, what we want to focus on sharing, and what ToB product managers need to plan, is the third category.

Third, what is the meaning of the existence of the SDK?

The answer is: efficiency.

SDK will be commonly used functions encapsulated in the form of components, so that other project developers, only need to simply integrate the SDK, call a few interfaces, you can use the functions that have been encapsulated in the SDK. No longer need to repeat the wheel, can greatly improve development efficiency.

Especially in the field of mobile games.

Many game products can be developed within a few months, an important reason is to be able to directly call payment components, voice components, security components, data communication components and other SDKs, developers only need to focus their core energy on the developers of game play. Greatly shorten the game development cycle.

Fourth, what are the demands of enterprise users for SDK products?

They want.

To be able to complete access in the shortest time, to be able to complete SDK verification in the simplest way, and to use the functions provided by the SDK. At the same time, during the operation of the SDK, there should be high enough stability and compatibility, and the impact on the performance of the APP should be small enough.

V. The existence form of SDK products

Most of the current ToB products are providing product functions for mobile products, especially Android products, so we take Android products as an example here.

An SDK product probably contains the following content.

1. SDK function module

In the field of mobile development, you can probably understand the following concepts: for native layer development with C/C++ language as the core, we will generally provide dynamic link library files in .so format; for jave layer development with java language as the core, we will generally provide library files in .jar type; for game development with C# language as the core, such as Unity, we will generally provide .cs format function module.

SDK function module contains the core function implementation. 2.

2. API interface

With SDK modules, programmers need to call these SDK modules to use the functions in them. And how to call it? API interfaces are functions that developers can load into their projects to use the functions in the SDK by calling these interfaces.

3. documentation

How to load the SDK module and how to call the API interface? What do you need to pay attention to in this? All these need to be clarified in the form of documentation to the developer.

4. Demo

Even with the above functional modules, API interfaces and documentation, developers using your SDK product is still very abstract, so it is best to give a project demo, the demo has a detailed sample code to explain how to use the SDK in the project, so that developers can understand at a glance, fast coding.

Six, the SDK product design 8 big tips

SDK development work is done by technical students, but they are more concerned about the implementation of the function. Product managers are responsible for the final product form of the SDK, product packaging, and then provided as a product for external companies to use.

So the product manager should have the most sensitive grasp of the usage scenarios of SDK products by external vendors, and then translate these concerns into product requirements, and integrate these user experience level requirements into the SDK development process of technical students.

So, what are the concerns for enterprise users' developers when they use SDK products?

What are the tricks for ToB product managers in planning SDK products? Let's summarize the top 8 tips for SDK product design.

1. The fewer the interfaces, the better

In the customer's product development process, SDK calls are only a small part of the process. Their developers' demand for SDK products is to complete the access in the shortest time and be able to use the functions provided by the SDK in a stable manner. So, how to ensure that the SDK call is simple. The way is that the fewer the interfaces, the better. It is better to call only 1-2 simple interfaces to complete the access, and the SDK access time should not exceed half a day. 2.

2. Have a simple demo

Providing a simple demo for the SDK has become a necessity for the SDK product. For the customer developer, when calling the SDK, referring to the runnable demo project provided to access the SDK, it will be half the effort and the success rate of access will be greatly improved.

3. Have clear documentation

A good SDK product is essential to have concise access documentation. The documentation needs to be clear and organized, describing the problems that developers may encounter in the process of accessing the SDK.

In terms of format, it is best to use markdown format, which is a structured form of documentation that can be used in a more standard uniform format or in a structured form of presentation when ported to the official website. 4.

4. the smaller the size, the better

Developers many times integrate 5 or even 10 or more SDKs, so if the size of each SDK is relatively large, it will eventually have too much impact on the size of their project or the size of the APP package. Therefore, this requires that when developing the SDK must pay attention to the size, streamline the code and functions, and provide the most core functions in the most streamlined form.

5. Fully adapt to various scenarios

SDK as a project component, it needs to adapt to a variety of project scenarios. Mobile development projects, for example, need to provide at least two types of Android, iOS SDK version. Take mobile game development as an example, it needs to adapt to various engine languages, such as the SDK needs to support Cocos engine, Unity engine, Unreal engine, etc. 6.

6. sufficient stability and compatibility

SDK as a third-party component is not very controllable for customers. They are not clear about the internal logic of the SDK, and may not change the logic of the SDK. Once access to their APP project, this is equivalent to the existence of a black box. Once this SDK crashes, it will endanger the operation of their APP. Therefore, they are very concerned about the stability and compatibility of the SDK.

To ensure the stability and compatibility of the SDK, two things need to be done.

First, when developing the SDK, we should make more efforts from the principle to ensure that the SDK is able to adapt to various models and scenarios in the basic mechanism.

Secondly, before launching the SDK to the public, we need to conduct comprehensive testing on various models and scenarios.

Here is another tip, before the customer uses the SDK, the developer can be reminded that he can control whether the SDK is running or not. It is convenient that if there is a problem, the SDK call can also be closed in time through the background switch.

7. access to self-test to be simple

SDK access is complete, the developer needs to verify the success of the SDK access. The most basic method is to provide developers with a tutorial, developers themselves check the logs and other methods to observe whether the access is successful.

There are two other methods that are more productized in Chengdu.

First, local access, Web real-time data verification

In the background, according to the data reported after the SDK access, the access detection results are presented to the developer in real time on the Web side. The developers only need to simply run the project project with SDK access locally.

Case: tdw.qq.com

Tencent Data Master SDK access acceptance function

Second, Web-integrated SDK installation

Some SDK products no longer allow developers to carry out complicated access, and directly require developers to upload the APP, and then unify the SDK into the APP, through the machine-based SDK input method, eliminating the possible misuse of human access, and eliminating the process of SDK access. It is simpler and more stable.

Example: gameguard.nprotect.com

8. Secure data

After the SDK access, it exists as a third-party black box in the customer's project. All companies will be very worried about whether the SDK has a back door and whether it will get all kinds of sensitive data to report. In addition, especially for projects running overseas, after the EU introduced the GDPR policy, there are stricter requirements on the right to collect data and the privacy of users, so it is important to pay attention to it.

For the SDK data security issue, the SDK product design process requires three points.

First, strict management of data collection, SDK in the development process, you need to do: do not collect the data you do not need, the data collected for business needs must be strictly encrypted to prevent data leakage.

Second, in the process of product promotion to dispel customer concerns. In the publicity, focus on presenting the SDK data permission issue, and promise customers the rigor of data collection and confidentiality of data storage. Eliminate customers' worries.

Third, remind customers of data privacy issues when they use the SDK. Your customers need to remind their users what data will be collected, and they need to get their consent before they can use the SDK product.

The above are the 8 major tips for SDK product design, which are of most concern to customers and most urgently needed for product design.

VII. Some SDK product cases of ToB category.

The following are examples of SDK products that can be used to compare and learn.

(1) AU: https://www.umeng.com/

Youmeng SDK download

(2) Gvoice: https://gcloud.qq.com/product/6

GVoice SDK download

(3) YouMi: https://www.youme.im/

YouMi SDK download

(4) Baidu Cloud SDK: https://cloud.baidu.com/

Baidu Cloud

(5) Tencent YSDK: https://open.tencent.com/

YSDK

Eight, summarize a sentence

SDK product form is an important part of ToB productization. Product managers can use the above 8 techniques in the process of SDK product design to launch highly productized SDK products together with their companies.