#### 腾讯--组件化如何实现，组件化与插件化的差别在哪里，该怎么选型

本专栏专注分享大型Bat面试知识，后续会持续更新，喜欢的话麻烦点击一个star

**面试官: 组件化如何实现，组件化与插件化的差别在哪里，该怎么选型**

**心理分析**：面试官从架构层次 了解求职者是否用过 模块化 组件化 和插件化，在过去经验有没有运用过这些技术到项目中，这道题属于一个连环炮。求职者该格外小心

**求职者:**应该从App开发的需求来定义技术选型，分别说说模块化，组件化 插件化的优势和区别

##### 一、组件化

组件化，就是把APP拆分成不同功能模块，形成独立组件，让宿主调用。 组件化不一定是插件化，组件化是一个更大的概念：把模块解耦，组件之间代码不依赖，宿主可以依赖组件；而插件化则具体到了技术点上，宿主通过 动态加载 来调用组件，宿主不依赖组件，达到 完全解耦 的目的（比如图片缓存就可以看成一个组件被多个 App 共用）**。**

**适合于项目大 但是功能相对集中**。比如 一个金融类的App 里面只包含金融的功能，金融功能又会有 借贷，理财，线下交易，把这些模块抽成单独的组件

##### 二、插件化

Android程序每次更新都要下载一个完整的apk，而很多时候软件只是更新了一个小功能而已，这样的话，就显得很麻烦。如果把android程序做成主程序+插件化的形式呢，这样才利于小功能的扩展（比如一般 App 的皮肤样式就可以看成一个插件）。

通过 gradle 配置的方式，将打 debug 包和 release 包分开。这样会有一个好处，开发一个模块，在 debug 的时候，可以打成一个 apk ，独立运行测试，可以完全独立于整个宿主 APP 的其他所有组件；待到要打 release 包的时候，再把这个模块作为一个 library ，打成 aar ，作为整个宿主 APP 的一部分。而 debug 和 release 的切换都是通过 gradle 配置，可以做到无缝切换。至于模块之间的跳转，可以用别名的方式，而不是用 Activity 和 Fragment 类名。这样所有的模块和宿主 APP 都是完全解耦的，彻底解决了并行开发的可能造成的交叉依赖等问题。

主要原理是：主要利用 Java ClassLoader 的原理，如 Android 的 DexClassLoader，可动态加载的内容包括 apk、dex、jar 等。如下

###### 插件化的优势：

* 适应并行开发，解耦各个模块，避免模块之间的交叉依赖，加快编译速度，从而提高并行开发效率。
* 满足产品随时上线的需求
* 修复因为我们对自己要求不严格而写出来的 bug。
* 插件化的结果：分为稳定的 release 版本和不稳定的 snapshot 版本，每个模块都高度解耦，没有交叉依赖，不会出现一个模块依赖了另一个模块，其中一个人改了这个模块的代码，对另一个模块造成影响。

淘宝的框架是用了osgi的bundle概念，整个应用框架生命周期完整。

**适合于项目超级大 但是功能相对不集中。**比如 一个支付宝App 里面即包含共享单车 也包含 电影票。这种与本业务完全不同的 可以做成插件的形式

###### 插件化弊端：

每一个插件都是一个apk，插件多的时候管理起来也麻烦。