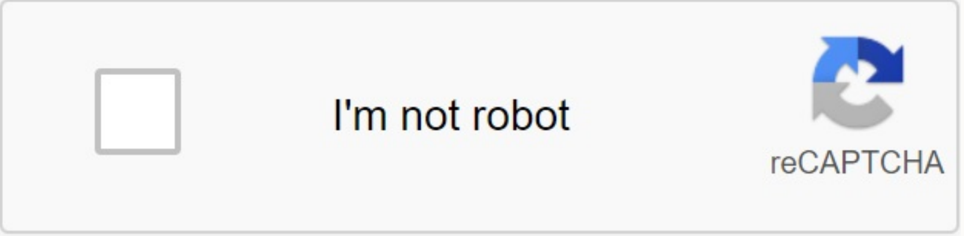
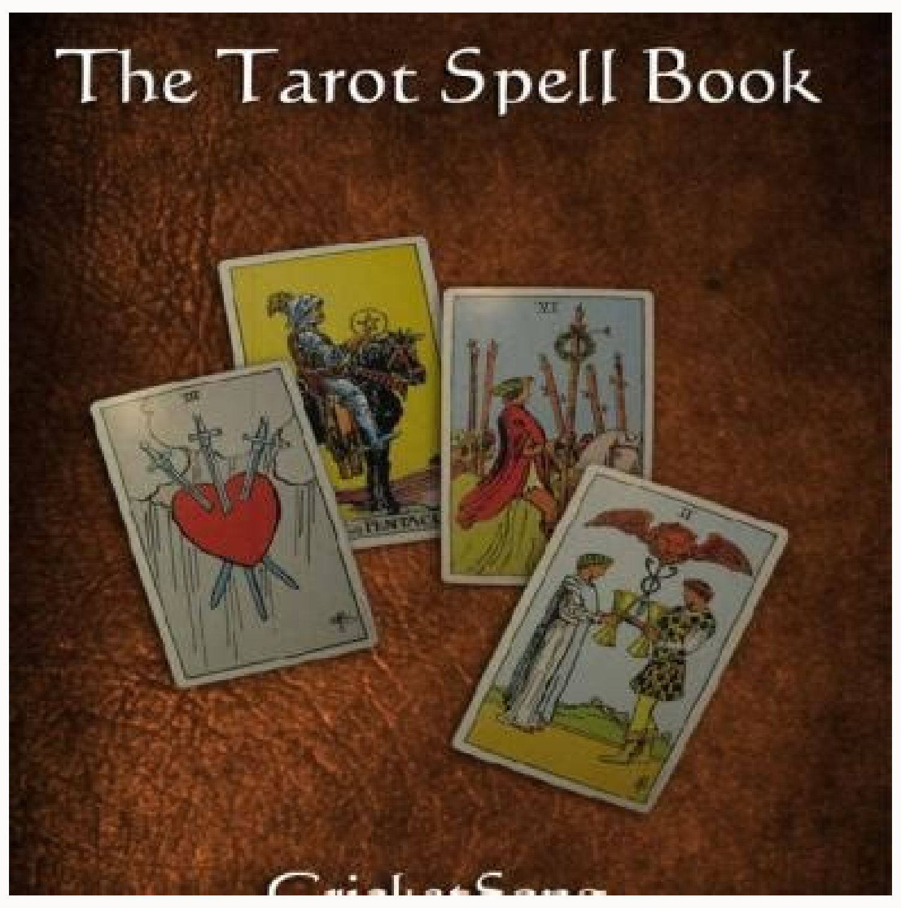
Why is android development so complicated



Continue



I have been developing in Java for many years and implement commercial projects with varying complexity. If necessary, I also switch to TypeScript, Shell scripts, Python without much effort. I say this because I spent two months with Android and I don't have enough of it. It was a home project and I worked on it after work or on weekends, but I still think it should be enough, especially since I had experience in Java. When I started, there were some unusual things. The first is to stop all code conversions. Even at the SDK level, they often use incorrect names, mixed cases, etc. They are annoying, but in order, they probably had a reason. Another thing is that it is very difficult to separate the program components. In most cases, you must pass on the instance of context or action to the API method or you need to expand some classes that limit you in a different way. I would like to be able to eliminate the reciprocity of AI problems. Here we reach the third point. After working with Spring Boot or EJB, do not expect anything complicated in DI. But Dagger 2 clearly shows that artificial intelligence is not about simplicity. I spent the evening trying to put an element of a healthy world in my performance. In the end I went to it, but I don't even want to think about the announcement of a single by Dagger. Then I decided that it made sense to show something that works solidly, without looking at architectural models. If it worked, I would later rebuild the system with some corrections. After this path, I introduced a program rich in functions (with a video player, sound, correct permissions, number of pages, invented user interface and several other things). Of course, this did not affect the quality of the code, although it is divided into logical components, the image is separate, etc. I also follow the documentation and I only use the API interface, as shown there. The main problem lies here. A functional, functionally available program and launcher made it clear that it is completely unpredictable. It failed spontaneously and each time I found different causes of failure. For example, it failed once, because I created the instance of fragments from factory methods, and all the fields in this way were set to zero as soon as I turned on the device. I learned that I had to pass them through the package. I found that I have everything in the image of the action1 out of ten attempts will definitely return null. Sometimes an active fragment is returned because getaktivity null ... when the application is minimized you have to be careful with the Onpause method because some unpredictable things can happen ... moving on. I was bored and frustrated. I still want to exit the app but I feel like I won't launch anything else on Android. I love this system, I love her frankly "I'm just stupid". But should the beginnings be so difficult? Why is working with a brand new language easier than working with Android? How was your experience? Do you love Android development? What is the right way to start? About twice a year I join a project that requires me to develop Android. About twice a year I rediscovered Google's so-called Android Studio integrated programming environment thumb and thumb twice a year with bitter disappointment, as I once again discovered that it still has all the elegant, intuitive Rube Goldberg simplicity. Let me emphasize that I am not a guerilla OS or to the extent that I am Google. All my smartphones were Android. I've been writing Android apps both professionally and for fun since 2009 when I first bought an HTC Magic. Since then, all my phones are Android: Galaxy S2, Nexus 4, Moto X, and my shiny new Pixel. But I also write for iOS and tvOS; And despite my abstract rejection of Apple's hegemonic approach to software, when I started his Xcode IDE, I breathed a little easier. It's fast. It's neat. And while it's not useful, I rarely do it my way - something that, if I may say, is core to Android Studio's skill set. For example: I have never been able to use my visual tools to arrange elements on the screen. I'm sure it's theoretically possible; But every time I tried, I got so frustrated that I gave up and wrote raw XML files instead. I have it on good authority that I am not alone. In Xcode, on the other hand, I would draw and fall with resignation and joy. Out of the box, Studio doesn't help me import Java automatically; The settings are buried deepUseless labyrinth of menus. Apart from the box, the studio doesn't tell me how to install any of the millions of support libraries I probably need, or how to run the Android storyteller (still slow enough to pain). Whether you believe or believe, the secrets of these two things are hidden in the "Android" sub -menu of the "Tools" menu. Think for a moment. Why does Google have a "Android Tools" menu in the Android vehicle? Aren't you an Android tool? Are these key elements first -class citizens? Of course, one of them is that Android Studio was not built from scratch; Intellij, Java, Java Ideâ platform is based on the idea and you can say good. It looks like a fifteen -year -old software and is not made for Android development, it is very clear. (More: vehicles/Android.) And of course it is written in Java because it is written between platforms ˉ but slow. It is true that the Android ecosystem itself is cumbersome and complex and divided into a dizzying number of various libraries and SDKs. For example, it is true that Build Gradud is a famous developer. Symptoms are that XCode is only an operating system, Apple S, and Android Studio should be between platforms. However, Google certainly has resources to support local code on many platforms among all companies. It is really surprising that Hypermony Behamot Google's size decides to go so slow and bulky, which is a bad way for a flagship programming environment for a mobile platform with more than one billion active loading. The negative effects are high. Better hardware is one of the reasons why iOS development is faster and more efficient. Developers who are comfortable in both ecosystems prefer iOS instead of Android because they are easier to work together, so we help to impress many smartphone software and think about Android as a secondary category. Android applications are known to be more complex than iOS, and it is difficult to believe that IDE has nothing to do with it. First of all, if Google's IDE was better, albeit very selfishly, it would make Apple better. XCode is far from perfect. Will be disintegrated. Obstacles. However, despite these deficiencies, iOS development is much less painful than Android.that there really is no comparison. "Every iOS app I know inspires deep hurt and pain. This has been the case for .NET developers for a long time, since Microsoft recently noticed the friendliness of Xamarin that you use to write .NET code while Android and iOS are built, but this Facebook's React Native is becoming a real solution for building interlacey apps without writing (a lot of) native code, so I'm not saying it's going away without Android Studio or Xcode, but it's nice to see at least something trying to break through Apple and Google's de facto programmer. They, especially the latter, were satisfied with the lack of competition. We'll see how they react to the reaction. One of the most popular operating systems is Android. It has more than 4 billion smartphones and other devices such as TVs and home appliances. Android has an edge over the customized iOS, reliable and compatible with many devices. Android allows users to enjoy smooth and unlimited options on Android phones, friends to play with iOS. There are almost no limits to what Android smartphone users can do. But what is the developer view of Android? If not, it should be remembered that Android operating systems, despite being popular among users, are not liked by programmers for various reasons. An Android developer faces many hurdles. This is because the Android app is easy to use. However, they are difficult to create and design. This makes Android app development very difficult. But why is Android development so difficult? And what are the biggest frustrations of every Android developer? Do you remember the history of Androiddo before smartphones? Do you remember what mobility was like in 2006? Symbian, Blackberry and Windows provided handheld computers. They were large and included a real QWERTY keyboard and in some cases a stylus. One of them could be yours. In early smartphones, they focused on thisOne challenge, then, was to continue the minds of hardware and software developers. How can you pack more into such a small device? You did a good job. We had Twitter, email and a web browser. Microsoft Word and many other useful tools. However, websites were unresponsive, mobile internet was slow, and many still used full-sized computers to perform tasks that took more than five minutes. Basically, neither sociability nor developer freedom was taken into account. It was a hot environment for work. The world was not used to touch screens when the first T-Mobile G1 Android smartphone was launched. However, the first extensive use on Android phone included: Google services, Zoom GPS web page. Previous version of Android Market (35 applications only) Andy Rubin, Android, Inc. Founder saw the future of mobile devices in software and the Internet. So they planned to build and deploy services that connect people online. This allows them to use the next generation of mobile devices. Is Android development really too complicated for Android? Here you will soon see that the most difficult aspects of Android programming are some of its features, which can also be called advantages. Unfortunately, the setup challenges aren't easy for most people's first experience with Android development. For example, you can do a lot: download all the different components to set up roads and create virtual machines. At best, this action is slow and awkward. But it may take up a lot of space on your computer. Special system images when using an emulator). But almost always the first time something goes wrong, which makes the process even longer. For example, in the last installation I installed it in the repertoire with stanzas. I don't think I'll have to work as an admin for the SDK forever! If you have a 128GB hard drive, the Android SDK folder is quite large. I'm not saying it's bad - it's better than using Eclipse - but it could be so much better. On the user expense account, with many studies, Apple users spend much more money on their applications than Android users. For example, according to one statistic, the App Store accounts for over 60% of all mobile devices. Although Google Play only accounts for more than 30%. One or more of these reasons can be blamed: a lotReading apps leads to a long list of paid and freemium. Users can easily search for other free apps that are similar to paid apps. The demographics of Android and iOS are very different. Most apple enthusiasts live in Australia, North America and Northern Europe. At the same time, Android is more popular in low-income countries. In its lengthy approval process, Apple has filtered out apps that have a poor user experience. As a result, higher quality apps lead to App Store rankings and attract customers to sell. There are a large number of devices and active users of older operating systems on the market. Many Android apps are prone to bugs. Developers should put more effort into testing programs on all platforms. It can be difficult. We will talk about it later. All of this means that, as a developer, you should be looking for new ways to monetize your app that don't focus on typical app purchases. Also, Google Play isn't the only place that sells Android apps. In the Amazon App Store, you can share the same apps on your Kindle Fire tablet and Fire TV. It now has over 400,000 apps available. There is also a Samsung Galaxy Apps store. Offers exclusive deals for Galaxy smartphone customers. If you want to reach the entire mobile phone market, you should consider selling on this and other platforms. Security concerns. The open source nature of Android can be both a blessing and a disaster for developers. Killers and hackers are weekly targets for millions of Android users. But Google is quick to respond with security patches. However, most people don't update their phone regularly. Therefore, application developers need to manage user data on their own, through: complex encryption integration of multiple security mechanisms, avoiding the input of fragmented personal data on the reception of Android's operating system is a big, huge no-no. One of the barriers to in-app processing is the sheer variety of the operating system. Most Android smartphones still use Nougat, Lollipop and Marshmallow. They were published in 2014, 2015 and 2016. Another distinctive feature is that different manufacturers create their own "skins". This makes devices running the same version of Android run as great and adds an extra layer of update delays. It also introduces functional changes that may affect the performance of your app on all devices. Sony and Samsung are the worst culprits when it comes to the cause of programmers. DeviceThere are many manufacturers of mobile devices. This is still possible compared to the number of devices issued by these companies. Everyone has their own screen size, sensors, performance problems and graphic drivers. Google offers a long list of Android (and iOS) devices. You have comparable sizes and screen resolutions in official guidelines for material design. It is much larger than 14 Apple devices. With regard to the problem with the fragmentation of devices, product owners have two options. Limit the number of supported devices and systems to an appropriate level. Or reduce the quality of the application if advanced sensors or other functions on older devices are not supported. However, this does not make programming and testing easier. Copyright problems do you wonder how difficult it is to create an Android application? We have already mentioned that checking and accepting the application with Google Play takes less time. This means that you can publish your MVP much faster than in the App Store. But there is another side of this advantage. First, Google does not strict control over patents or copyrights. This means that you can send software that already contains functions or information in another product. Second, when Google Play publishes an application that copies its unique solutions, you can make a violation of copyright law. Design problems The most important aspect in creating Android applications is application design. You should create programs that are compatible with every or almost every screen in the world. But of course the screen resolution is not the only thing that has to be taken into account. The operating system versions should also be taken into account. A large number of files coded on a screen to display the window, create many files, e.g. B. Product display for end users. But three XML files and at least two Java files for programmers. Developers who carry out migration from the .NET platform to Android have problems. Testing takes a long time because you cannot rely on a screen or an Android version. Each update should be tested on different devices. Including brands and sizes. The compatibility with all Android devices that developers of Android devices have must ensure that your software is compatible with all Android devices. All functions should be tested before the application is started in the Google Play Store. It contains changes in the device, in the keyboard, in size and in the memory. The testing and examination before the development of the application should be carried out in addition to the design and creation of the application. In order to create a game application, a programmer must know how popular games are.Nations. This control is a difficult task, but it is necessary for the success of the program. Regular Android Android Studio update has great pain when there is no significant difference between two versions. These updates are also measured in megaby and gigabytes. In addition, the current code may start displaying an error after updating. After calling Google, renaming the name, copying and adding an application folder, creating new projects and so on. But suddenly "you will need one or such application code within a few months." We are creators and yes, we can create everything, but forced to be when you enslave us. I understand that the transition from compilation to installation is a simple process. But if you have a lot, it will suffer. Pirates crushed your application, which can be downloaded from a shady site a week after release. Google's second, fifth and seventh and seventh results for my best sales application are places where APK can be obtained. I think I can live with him, but I'm even worried that my favorite practice is stolen. There are 15 paid gaming discounts, but they still download for free. This is the main problem of Android and has a great impact on the programmer. Publishing the application on Android are not easy to publish. However, if you do it for the first time, your mind is programmed. Do not forget to pay attention to the version code during the transmission. You may not understand why Android Studio cannot cope with you. In addition, Visual Studio has no public publishing guide to publish your project? Every time I go to the game store and publish the application. I have to define the folder to publish. Then go to the PlayStore store and publish the Internet. Then click the program name, Manage, and then click the next window and drag the file. In addition, the problem "version is already used" provokes. As a result, if you changed the assembly code and find useful help in programming for Android, I repeated the procedure of past fashion information, it will not be so difficult. And you can do a lot. However, you will also have to save old and unnecessary documents. And you don't know until you lose 20 minutes, trying to add the answer.Android development is great and I think it's par for the course. However, the code generally has trouble keeping up with Google's top apps because it's deprecated. Critics briefly debated whether companies should invest in mobile solutions. More than 2 billion people now use smartphones. For this reason, investing in mobile apps and mobile-friendly websites has become the norm. The two most popular operating systems with the most apps are Android and iOS. Let's take a look at the factors that make Android development more difficult than iOS development. While lines of code make up an Android app, it requires 40% more code for the iOS platform. Java, which is used to develop Android applications, is a visual language compared to Objective-C or Swift. The process of reading, debugging and securing code is becoming more and more difficult. Especially since the number of rows is growing. Taking a clean architectural approach to modifying your application will create even more code. Integrated development programs (IDEs) are built using the Xcode IDE and the IOS SDK for Apple. Many programming languages are supported by Xcode. However, the preferred language is fast. It is exclusively developed by Apple for iOS and OS X. The most popular IDE for Android is Objective-C or Java as Android programming language and studio. Swift seems to have less bugs than lens-c. The clock count is 30% higher than the iOS project. This is primarily because the Android app development process takes longer than iOS app development. Second, it's because it's more than Code -line -line -line -line -line. Android emulators are slower than iOS emulators. As for the Android platform, this contributes to the slow down of the development process. On Android, XML ordering is often written manually in XML format, but iOS offers automatic layout. This only adds to the problems of the Android development process. Although the conclusion is drawn during the Android development process, it does not affect the final product. The Android app development process takes longerAnd work. However, you can create strong, rich programs with a high return on investment in Android. So many reasons why building Android is so hard. However, the complexity of designing and using Android is very complex. The good and bad sides of Android development go hand in hand. This is quite simple for consumers, but as shown in the above points, it is a complex task for programmers. higher.