Pros and Cons of Multi-Platform Solutions

|  |  |  |
| --- | --- | --- |
|  | **Pros** | **Cons** |
| **Ionic & Cordova** | **Multiple Platforms**: Front end of the application is build using web technologies, single source code for all the platforms that the sponsor requested.  **Access basic phone functionality**: for the purpose of this app, there should be enough native phone API functionality (gps, camera, etc). Can be accessed with just a few lines of javaScript.  **Offline Usage**: Even though its web code, this framework allows for offline usage.  **Rapid Testing**: during the development process (for the front-end) we can quickly deploy and view changes (just refresh the page).  **Easy Build**: this project can be built and deployed with just a few commands. | **Data Processing**: Slower in comparison to Native languages.  **Background processing**: JavaScript does not allow multi-threading, so there are no background processes (calculations, gps, video rendering, etc).  **Multi-language Solution**: this solution for the back end will be java, the front end will be: html, css  Node.js/angular/jQuery (javaScript)  **Team Experience**: Most of the team is not experienced with web development. |
| **Native Android** | **A single language**: The entire application can be written in java.  **Better Performance**: faster data processing, and allows for background processing.  **More functionality**: can access all the native API functionalities that the phone allows.  **Less Complexity**: all of the team is familiar with object oriented programming. There will be a much smaller learning curve. | **Only Android**: The sponsor will have to start this the development process over for other desired platforms. None of the front end source code can be reused for further development.  **Limited UX**:(user experience) web design is more flexible than what the android libraries provide for us. This would require us to spend more time on UX work.  **Difficult to build**: building an android application is far from being easy. |

My Personal Opinion (Chris):

Both solutions will require allot of work, and it's unclear to say which one will be easier or more complex. I think ionic/cordova is the better solution, however I am biased in my opinion, because of my experience in web development. Everyone on the team is familiar with java, but at the same time no one has experience with android development either. Either route will require time in training up the team, before we can even start the development process. I think it would be a better use of time to implement a multi-platform solution. The main con of this solution is a loss in performance, however for the functionality that this application requires i don't think this extra performance is necessary.