

Why GRENADES are a Better Choice for Mobile Application Development

Executive Summary

There are a number of vendors that are singing a “siren song” about their product to management leadership and developers across the marketplace. The following discussion describes a number of reasons that those who listen to that song can find themselves shipwrecked in their pursuit of what at first blush looks like a good approach to mobile application development but ultimately ends up dooming them to the reefs of mediocrity.

Terms

GRENADE: Generally REcommended Native Application Development Environment
For iOS this is Swift and Xcode. For Android this is Java or possibly derivative languages (Scala, Kotlin, etc.) and a suitable toolchain.

MALMADE: Meta-Application Language in the Middle Application Development Environment.
This is a general term for third party products which typically use some well known meta-language (like JavaScript) and then convert/compile that into the respective native languages/bytecodes for the target platforms..

Background

I’ve been an application developer on all kinds of computing equipment for over 50 years. I’ve been doing mobile application development since the first WAP (Web Application Protocol) enabled phones hit the market back in 1999. The next big thing was the emergence of J2ME (Java 2 Micro Edition) back in 2006. When Apple introduced the iPhone back in 2007, we had a platform that allowed true web applications to be developed. They were awesome and easy to develop compared to J2ME but still just a webpage served up by the smartphone. Developers wanted to do “native” development to unleash the full power and potential of the devices. Apple launched a developer “beta” program in February of 2008 and I was a charter developer. The first app store opened in July of that year. My first application was published therein in August. Many more have followed including a number that were “ghost written” by me for other companies as a consultant. Suffice it to say that I’m about as experienced as you will find in this genre.

The Issues

The primary issues stem from the market. Apple is arguably the single most dominant player in the smartphone market. Google and others using Android are probably the main competition. I didn’t mention the Blackberry which used to rule that roost. It seems clear that they are pretty much a niche player vis-à-vis the other two.

As a result, enterprises that want to get into mobile development generally want to be able to address the two primary markets if they are serving the general public. If they are developing primarily for enterprise use, then the enterprise will focus on the platform adopted by the enterprise.

Mobile developers face this same issue. Which platform should I learn? Is it practical to try and learn both? What about one of those products that let me learn one thing but publish to both? The answer to the last two questions is (in my opinion) a resounding NO.

The Pace of Mobile Technology

In my five plus decades as a developer, I can't think of any technology I've experienced that has evolved as fast as mobile technology. From the beginning it has been clear that, short of an almost fanatical devotion, it would be impossible to keep abreast of both of the iOS and Android platforms. I wanted to make money, so I chose iOS as it had about a two-year head start in the market vis-à-vis Android. Today, you could choose either and make money although Apple still has a bit of an edge in terms of realizing the potential.

Now imagine the plight of a company trying to build a MALMADE product. They not only have to try to keep up with the two technologies, but they have to try and integrate that into their own product. In truth, they simply cannot keep up. They will always lag significantly in terms of available feature/function as the underlying platforms evolve. So as a developer for an enterprise, if I'm happy to have my product always lagging significantly in the state of the art the MALMADE approach might fit.

The pace of the Apple platform has been incredible, including a transition from Objective-C as the primary development language to Swift. From a somewhat more distant perspective, it seems as though Android has evolved at a similar pace.

Obviously, the GRENADE approach allows developers to leverage the state of the art. MALMADE developers have to wait for their MALMADE vendor to incorporate and integrate the improvements from all of the native platforms they support. That is a daunting task. Because of that there is either serious lag in available functionality or outright absence of key features that are readily available to native developers.

Write Once, Run Anywhere

This is the essence of the "siren song" the MALMADE vendors sing. The appeal to the individual is that he/she can learn one meta-language and cover BOTH major mobile environments in the bargain. However, the sheer number of differences in the underlying technology means that any such product by definition HAS to produce a "least common denominator" product. Java apps are a case in point in the PC world. Most folks on Mac, Linux, or Windows platforms prefer native applications over their Java counterparts. This is even more apparent in the mobile world as there are far more differences between the mobile platforms than there are in PC platforms.

For enterprise management, the “siren song” has a different verse. The pitch is that you only have to have one skillset on your staff for the MALMADE approach whereas the GRENADE approach requires maintaining/training two skillsets. Of course, this is moot if you are only doing enterprise development for one or the other for whichever device the enterprise selected to adopt. Even then, most enterprises have some BYOD (Bring Your Own Device) users they wish to satisfy in reality. I have yet to see a company that is “pure” in terms of having a single mobile technology client base.

A More Excellent Way

It turns out that the costliest part of mobile app development is the initial development. I don’t mean the coding. Coding is easy. It’s getting the design specifications nailed down. Customers know they want a mobile app but when you ask them what they want you typically get the classic “I’m not sure but I’ll know it when I see it” response. The net outcome is that you invariably go through a lot of iteration with a customer because when they get the first prototype, they will immediately “see” what is lacking in terms of their expectations. You would think that they could have said what they want up front but in over a decade of doing this kind of development, it just doesn’t work that way. So rapid iteration is key to keeping costs down.

I’ve found that the key to low cost development is to quickly get the application defined in some platform and get it into the hands of your client/customer for evaluation so that you can actually nail down the appearance, user experience, and function that meets the need of the customer. That’s the hard part and it pretty much doesn’t make a difference which platform (iOS, Android) you start with as long as you are competent in it and you are able to develop quickly and get it on the customers device for feedback. Babcock’s tenth law applies: “It’s easier to get criticism than criteria.” https://www.babcock.org/is_laws.htm

As a consultant, one of the first questions I get asked in an engagement is “Can you do an Android version in addition to the iOS version?” I always answer “Yes, right after we do the iOS version to your satisfaction.” It turns out that merely duplicating the function of an app on another platform is straightforward for any developer competent on that platform. That’s because it’s merely a matter of copying the user interface (with appropriate form factor adaptations) and the user experience (with adaptations suitable for the target device) and overall function. Copying is relatively easy. Just ask the Chinese!

I believe it is far more effective to develop a given application on either iOS or Android using the GRENADE for the chosen platform and then simply outsource the other platform version to any of a multitude of shops that specialize in development for the other platform.

It seems clear that an iOS developer skilled with the Apple GRENADE (Xcode) can develop iOS applications more effectively than someone using a third-party tool. The same argument would apply for Android development. In short, I believe the fastest, cheapest way to get to a satisfied customer is getting the app done in one platform and then merely duplicating it in the other.

One more thing. Remember that the cost is all labor for the most part. Those using MALMADE tools are doomed to pay that cost for both platforms instead of one. That's because they are essentially iterating over TWO platforms in parallel while the customers are making changes. In contrast, if you "get it right" on one platform, the other is merely a matter of cloning the feature/function.

An additional benefit of this approach is that you can easily tailor/adapt the function to best fit the full offering of the device/OS in each. When using MALMADE development, you typically don't have the option. You have to stay on ground common to both platforms. It's Least Common Denominator.

Support

In mobile application development there is all kinds of support on GitHub, blogs, and developer resource sites (like RayWenderlich.com a.k.a. kodeco.com) for the GRENADE development platforms. When I have a feature or function I need, and I don't already know how to do it, I can almost always find a wealth of examples from which to get ideas or just plain use outright. In contrast, the resources for those using one of the "me too" MALMADE toolchains are relatively scarce. If I had to hazard a guess, I estimate that web resources for users of GRENADE's is at least two or three orders of magnitude greater than for those using MALMADE's. That ends up being HUGE in terms of development labor costs. What's easier, dropping in a ready-made component that some other developer has perfected and published or coding my own? A few minutes of mining the web has saved me literally days of development effort. You simply will not find that kind of wealth for MALMADE's when compared to what you will find for those doing GRENADE application development.

For this reason alone, I'm quite sure that the TCO (Total Cost of Ownership) for application development is far lower for enterprises that use GRENADE's vs MALMADE's.

Recommendations for Management

- RUN, do not walk, away from vendors promising you a “silver bullet” solution in their MALMADE offering that will get you into the mobile application development game.
- If building up your own internal development resources, pick the platform that fits your enterprise. For example, if you buy your employees iPhones and iPads for corporate use, then it makes sense to focus on iOS. If you are buying Android devices, focus on Android. If you already have talent on board using GRENADE’s and demonstrating success, start with that platform if there are no other factors.
- Become competent in a single platform and simply outsource the need for the other if you don’t want to staff for it. Whatever you do, don’t develop in parallel or you’ll pay twice as much because you will be making iterative “not quite there yet” changes in multiple platforms instead of one.
- If you DO outsource, make sure they use GRENADE development for the version you don’t develop in-house. If they are using a MALMADE for development, you just inherited their sins and limitations.
- If you DO staff for both, assign development of a given project to one platform and then copy it in the other platform group to avoid doubling the iteration cost. Spread out the joys of creativity by using “round robin” or some other rationale for which of your two groups gets to lead on a given mobile project if there are no other factors. If you don’t have enough mobile application development for that, then you definitely should major on one platform and merely outsource the other “copy” version.
- Always use GRENADE’s for application development to minimize your development and support costs.
- By using GRENADE’s, you avoid vendor lock-in and its attendant disadvantages. You will be able to leverage any improvements in feature/function released by Apple/Google immediately. Users of MALMADE’s have to wait for those newly available functions to be included in the capability of the MALMADE product. That could take weeks and months if the MALMADE vendor decides to include the new feature at all.
- In terms of staffing, there are far more qualified developers available that have the GRENADE skillsets in either platform than those using MALMADEs.
- The corollary for developers is that if your skillset in the mobile world is only via MALMADE’s then you are pigeon-holing yourself in their niche employment market.