

COLUMNS

How to design for Google's Android

February 22, 2011



By AN AMERICAN MARKETER COLUMNIST

By Matt Joe and Tom Moran

Anyone on the agency side can attest to the fact that last year saw the application movement reach a fever pitch.

From Android to iPhone to iPad and Windows Phone 7, our cup continues to runneth over with requests from clients for applications of every size and shape.

But the volume and pace we have experienced when designing for Apple products is nothing compared to what is coming with Google's Android.

Apple will continue to do first-mover innovative work, and will continue to own the high end of the market for mobile devices, but the broad action in the mobile space will come from Android.



In 2009, Gartner predicted that Android would become the No. 2 mobile operating system in 2012. It seems they have already blown past that prediction.

Here are four observations we have made the hard way in designing for the Android environment:

Form factors and operating systems: Infinite options, infinite headaches

Even though iPhone and iPad application development is getting more complicated with the release of new iOS and hardware versions, we are not facing an avalanche of form factors, devices and versions of operating systems like we have in the past.

Remember doing Windows Mobile development? It was fairly common to deal with testing configurations of 15 devices against four versions of the operating system.

While we can say that, so far, Android development has not been like Windows Mobile of the past, just wait. It is about to get a whole lot more complex, but ultimately more exciting for native application development.

It is worth noting that Mark Zuckerberg of Facebook recently announced that the company does not have any plans to develop an iPad application.

This is a good reminder for the industry that a strategy of "apps for the sake of apps" on a platform is not always a sound long-term approach.

For example, if you are looking for the most bang for your buck in terms of achieving broad reach, building a native application aimed at one particular platform may not make the most sense depending on your target audience.

What may make more sense is to take a hybrid approach where you host as much as possible on the server for reuse in other platform applications and create a native shell for the hooks into the platform.

Or you can build an HTML5 mobile-optimized Web application, giving you a much bigger audience.

There are, of course, legitimate reasons for building native applications such as monetization, being a part of a trend and better performance.

If we do not help our clients think through Web versus hybrid versus native and supported device configurations and simply start with native and a platform broadly versus narrowly supported OS versions, we will find ourselves on a roller coaster of emotions and budget expansions.

After weighing the pros and cons and deciding to go with a native approach, focus on the platform.

When we design and build mobile experiences, we want only the best only what delights our users.

The knowledge that we have of the underlying platform becomes a differentiating advantage for an exceptional experience. Just be prepared to pop a few painkillers to achieve your experience for native applications if you do not limit the scope of support Android configurations, which can, unfortunately, limit your audience reach.

Those nice little comps won't cut it anymore

We digital marketing natives have spent much of our careers mocking up ideas in Photoshop quite successfully.

Clients get to see what we are dreaming of and see the processes and workflows behind the vision. Those days are long gone now that we are living in a mobile application universe.

The linear "backwards and forwards" experience that was once "enough" is now uninspiring to clients, at best, and misleading, at worst. It often leaves too much to the imagination. And in this context, imagination can be your best friend, or your worst enemy. The solutions?

Good old-fashioned prototyping, giving the comps the animation and interactivity they need for the client to truly embrace the vision.

Creating a light-weight proof-of-concept gives us the opportunity to refine and polish a concept to perfection.

Also, "Motion Studies" are becoming a powerful tool, along with static comps, to show how screens get rendered and how data moves or becomes animated.

Sound painful? Well, it can be. After all, you could be building something that may or may not get the green light.

What is more, you have to involve everyone from UX, to design, to development and QA from the very beginning to educate them on the concept and achieve full conceptual buy-in if you hope to have flawless execution.

Though an investment, it is truly worth it to avoid clients becoming dissatisfied when they finally see what the content, interaction, and movement look like together far later in the development cycle.

Consistent hardware and OS interfaces matter

With the iPhone and iPad, carriers are not allowed to create their own flavor of the operating system and add their unique user experience to the device.

As with past Windows Mobile devices, carriers are now actively creating or, at least, looking into creating their own Android experiences to create non-standard interfaces.

For those of us building great experiences, we know that designing for a known OS interface is one thing that helps guarantee this.

Consistency of OS interface is one of the reasons why we like developing for Apple products.

Not knowing what the carrier is going to do next strikes fear in the hearts of user-centric application developers.

We want to know that the experiences we put on devices will be consistently accessible, vibrant, relevant, and

we want to know that the experiences we put on devices will be consistently accessible, vibrant, relevant, and memorable.

End-users will be the winners if carriers and platform owners can find common ground.

Beyond the OS interface, the networks need to be fast and reliable, the hardware has to be of the highest quality, and the minimum requirements for the operating systems need to be enforced and generally high.

We hope Google can help carriers and OEMs find a good balance between their goals and the Android experience.

Until then carriers, we beg you: do not be penny wise and pound foolish with hardware.

Microsoft learned this lesson and is actively enforcing hardware requirements with its new Windows Phone platform. Experience matters to your customers.

We are in control

Google does what Google wants, but it does listen to the Android community. This is a blessing and a curse.

If we want something to change, we can take leadership roles in the community and drive an idea forward.

If we do not participate, we miss out on the dialogue and the community can work against us.

Do not ignore the community aspect of Android development.

Specifically, share code early and often.

If you build something that quickly becomes "baseline" functionality for you, get it out into the Android community so that the platform will continue to get stronger faster.

With our limited budgets, we should seek to spend more of our dollars on differentiated innovation than maintaining commodity logic, so give back.

THERE ARE MANY other details and observations but overall, it is a great time to be in the mobile industry and, specifically, to be taking on the Android platform.

Agencies, clients, carriers, device manufacturers and Google are learning side by side and innovating at breakneck speed.

So, while there is the potential for pain now and in the future, there is nothing here that we cannot handle.

The market is growing like crazy, the industry is fiercely competitive, and we are only just getting started.

Matt Joe is vice president of technology and Tom Moran is executive creative director at POP, Seattle. Reach them at matt.joe@pop.us and tom.moran@pop.us.

© 2020 Napean LLC. All rights reserved.

American Marketer is published each business day. Thank you for reading us. Your **feedback** is welcome.