

Spend Matters™

RESEARCH

Mobile Apps: A Passing Fad? Or a Procurement Solution Must-Have?

By: Thomas Kase, VP Research

This content originally appeared on Spend Matters PRO. It has been reprinted with permission for Vroozi. You can read the original piece here.

Introduction

Who wouldn't want to access their procurement solutions on mobile devices – whether smartphones or tablets, powered by some flavor of Android or Apple's iOS? If online banking can be done on a smartphone, why not e-procurement? The question is how much of the desktop experience can (or even should be) be retained on a mobile device? Running complex what if optimization scenarios on a sub-5-inch screen is probably not workable, but managing contract approvals, requisition and PO workflows should be easy to handle. So how does this impact the solution design, and how much weight should you (the buyer) assign to the mobile experience when you select your next procurement solution?

In my case, when I talk with solution providers, at some point we discuss what drives their design and how they arrived at their user interface and user experience (UI/UX) approach. Before I go into my own predictions and recommendations, just to make sure we're all on the same page, let me bring up some common questions and the technology reasoning behind them.

Is the solution HTML5 compatible? This means that the solution will work on just about any reasonably modern piece of equipment, regardless of whether it is a Windows or Mac laptop, an iOS or Android device and so on, with other benefits being an ability to include media without additional plugins (which corporate users might be tripped up by), flexible images and a flexible layout that implements relative sizing (to fit different size displays and rotation).

Is the solution driven by responsive design? This is a bit of a trick question since, if it was developed in HTML5 per above, it ought to be responsive, but you could create a responsive design website in HTML4, CSS3 or probably other standards too, but why would you since HTML5 has other advantages too.

Does the solution feature adaptive design? It is a little more unusual to get a "yes" to this question from solution providers – think of this as a higher-end version of the responsive design. For example,

instead of experiencing the full website on a small screen device, the web server provides those visitors with a preconfigured, reduced, experience that caters to the primary use case for the smaller device (for example approvals). The drawback here is the reduced functionality for smaller screens, but if done properly it can really speed up productivity through focused workflow and rapid load time. Consider adaptive design to be a close cousin to responsive design – responsive flows content to a given screen size whereas adaptive responds to a set of predetermined set of viewports, say a specific iPhone landscape view.

Do you have any mobile apps? And this is where some solution providers proudly declare that, yes, they have app(s) for approvals (the most common use case) for their solutions, or maybe some other use case that has an offline expense reporting solution. Yet others say, "No, we don't think that is worth our investment."

Along with HTML5, JavaScript and CSS3 allow modern browsers to interface and communicate with backend systems more seamlessly than earlier iterations – server to client communications can now happen without page refreshes using Asynchronous JavaScript (AJAX) calls and return notated data, either in XML or as structured JavaScript.

Mobile apps on the other hand are able to do things that a web browser cannot – for instance, they can store more data (to a local database built into the app) on the local device for offline access for instance (although you can store some data with HTML).

Why are mobile apps so expensive to develop? Consider the fragmented infrastructure of our computing devices – from super high resolution (500ppi or higher) screens all the way down to the tiny Apple Watch screen, some over 10-inches in size, others miniscule. Add in the various generations of iOS, Android, MSFT and other mobile operating systems and it's easy to see how it can be difficult to develop for all. There are obviously third-party developers that develop for and guarantee compatibility with the most common platforms, but it does increase development costs. Instead, consider the responsive/adaptive HTML5 compliant designs – here you build it once, and it'll work on anything.

Another reason why mobile apps are often more expensive to develop is because they access your backend data differently than the web, and they can be more complex than just getting a push of data to be read. They are also often an afterthought. Mobile apps are not restricted by the functionality of the web browser and can tap into system-level functionality (e.g. local storage, camera use) in ways that web-based applications typically cannot. For these reasons, for example, Vroozi has chosen to develop their procurement platform using HTML5 with responsive design.

Because of the expense most developers like to focus on getting the user experience right, so they can build on that and offer a unique experience with their product instead of cobbling something together to change it all on the next OS iteration.

Crowding out effect/too many apps – I've been an early adopter of smart phones - started out with proprietary Japanese smartphones (Toshiba J-Phone) years before anything like that was (commonly) available in the US. From the Treo 650 to early iPhones to Android devices, I've used a few. And currently I'm finding that I am loath to install apps that aren't truly critical. I periodically purge my

phone of apps, because it's just too much to deal with, and even mobile apps consume considerable amounts of precious memory. In fact, I even have an app to monitor which apps I don't use so I can uninstall those.

And in a corporate environment, would you rather install a few dozen apps to get your daily work done, or just access HTML5-compliant websites that load quickly and don't require any additional plugins or any other software? I know I'd prefer fewer apps. Count me among the skeptical when it comes to the future of apps.

I do see a continued use for apps in areas where you really need to engage with local data (tools where you might scan in receipts while up in the air or in remote sites, or travel tools like Delta's app that give you a copy of your boarding pass), or supplier performance management (SPM) tools where users are often onsite in locations either wholly without IT access (oil & gas, for example) or in locations with restrictive IT policies against external access to WiFi (data centers, financial services or client/supplier offices). Outside of those special use cases, I expect apps that add no functionality beyond manipulating online content to go away and be replaced by adaptive/responsive designs.

Ubiquitous WiFi – this is another trend that runs against the apps and their offline data benefits. We're surrounded by open WiFi networks, I've written in previous articles how even in places like Vietnam, I could find good quality WiFi connectivity, so why would you need an app if you can access the live data?

My recommendation for solution providers is to focus more on the benefits of adaptive/responsive – ensure that the solution can be accessed elegantly over a regular browser on as many varieties of screen sizes as possible. This has to be the primary focus. If you have unique functionalities (some listed above) those might be better served via an app, but then again, they might also be better delivered via adaptive design.

For practitioners, I suggest that the adaptive/responsive design is the one fraught with the fewest headaches – especially when considering that app installations on corporate devices are likely to fall under restrictive IT policies that try to both reduce the app count as well as QA apps before they are permitted, all of which is a lot harder than adaptive/responsive designs and their ease of deployment.