

Looking for new ways to present material
and reach more people in a digital format?

A MOBILE APP CAN BE YOUR ANSWER



You may not copy, distribute, transmit, email, reproduce, publish, post on a website,
license, create derivative works from, transfer or sell this electronic file.
© 2019 QuickSeries Publishing - September 6, 2019



A non-technical guide on how to **approach application development** for your community organization

By Erika Szucs, Editor in Chief, QuickSeries®

If you are looking for new ways to present material and reach more people in a digital format, a mobile application (or “app”) can be a natural progression and positive addition to the outreach your organization provides. But if you aren’t familiar with the process or the technology behind it all, the next steps can seem overwhelming.

With the right strategy, mobile apps can increase your organization’s visibility, while also helping to provide value to your external stakeholders. At the same time, if you adopt the wrong mobile app development strategy or employ the wrong technology, you risk negative publicity for your organization – and a lot of wasted resources.

QuickTerm

A **mobile app** is any native or hybrid (downloadable), or web application designed to be accessed and used on a handheld mobile device (e.g., smartphone, tablet).



What to consider when developing your mobile app from scratch

Building a mobile app can cost more to maintain than a website because, depending on the type of app you create, you may have to build the same thing, multiple times for different platforms – at least for iOS (e.g., iPhone, iPad) and Android (smartphones and tablets that run on an operating system developed by Google).

Why not just go with a mobile website? A mobile website is essentially the same website you would see on a desktop, but it has a “responsive design.” This means the website can scale itself to appear and function better on small screens as well as large screens.

Mobile apps appear to work much faster than mobile websites and allow for better interactivity with the user. Mobile apps can also perform more sophisticated functions for the user and are easier to personalize.



Types of mobile apps

Native apps are developed for use on a particular platform and use the native language of that platform (such as Google's Java-based "Android" or Apple's Swift "iOS"). These apps can access all the device's features (e.g., camera, GPS) and can be built to function offline. They can be expensive to build, test and support, and are often subject to app store approval processes, guidelines, delays, etc.

Web apps are accessed through the internet on mobile browsers and are portable across multiple mobile platforms. They generally have simpler functionality than a native app. Think of a website that's been optimized for a smaller screen. They are developed with web technologies such as HTML, CSS and JavaScript. They have limited access to a device's features. Temporary data storage called "browser caching" or local storage is available for offline use, but it's still limited compared to what you get with a native app.

Progressive web apps are built using web technology. They are essentially web pages – but with the new standards, developers can make them behave more like native apps, with improved access to a device's features. They do not need to be distributed through an app store or installed on a device as they can be linked and shared as web pages. They can be added to a device's home screen to make them easier to launch. However, not all browsers and devices are presently supported. At the time of writing this paper, iOS 11.2 and MacOS 10.13.3 do not support any progressive web app features (but features are starting to creep in as of iOS 11.3 and MacOS 10.13.4).

Hybrid apps combine elements of both web and native apps. The major difference between a native and hybrid app is the technology used to build the app, and the perceived performance you get out of animations and the rendering of the interface. Hybrid apps do not need internet access to function. Hybrid apps are usually built using web technologies, but wrapped in a native shell that gets distributed to, and downloaded from an app store, just like a native app. Also, hybrid frameworks are usually a little behind native apps since developers have to build bridges to use the new technologies available in new operating system releases (whereas native apps have access to them immediately).

Mobile app best practices



You will want your mobile app to be **fast, stable and offer a good user experience (UX)**.



You will need to **keep it up to date** as it will have to work with updated operating systems and new application programming interfaces (APIs).



While helping to expand what's possible to do on a mobile device, these ever-evolving operating systems mean that mobile apps **require a lot of maintenance and updating**.



Your mobile app will have to **stay current enough** to work on new phones and tablets, while still supporting older devices, too. For example, Android developers need to make sure their apps “support” – i.e., are tested and approved to work on – more than 15,000 different devices. The reason? Android is an open-source platform, which means that any original equipment manufacturer (OEM), e.g., Samsung, LG, Huawei, can “fork” the original project – that is, modify it to meet their own needs. An app that functions perfectly well on one OEM's device may stop working on another smartphone because that OEM runs a forked version of Android, which adds functionality but sometimes causes apps to crash.



Equally important to having a robustly designed app is having a **well-defined purpose and knowing your audience**.





Great mobile apps require intentional planning throughout the development process

To most people, an app is an app. Although they may be able to judge the good from the bad, they may be less able to pinpoint the characteristics that make it one or the other.

Many organizations struggle with their mobile strategy. One easy way to fail is thinking of your organization first – and not the end user. For example, users will get new devices and expect more interactive content, such as video.



Think about what information, features and functions will be useful for the people who will use your mobile app. As you move the project forward, you will need to ask yourself some basic questions.

Start by clearly identifying your ideas, goals and what the mobile app should accomplish. "An app should not aim to do 20 different things. Instead, an app should focus on solving a subset of similar problems so as not to confuse users with the app's main intent," said Justin Ledoux, vice-president of software development at QuickSeries®. For example, Airbnb originally chose to focus only on making it easier for people to find, inquire about and book places to stay. Staying true to their original concept, they have not added features for booking cars, activities or organized trips.

Here are some fundamental questions you need to consider before you start creating an app:

- 1 What is the overall objective or mission of the app?
- 2 What community or end-user need are you trying to deal with?
- 3 Who is your target audience:
 - a. What is the age group of the intended users?
 - b. What is their overall familiarity with mobile apps and digital technology?
 - c. Are there multiple audience demographics?
- 4 Will this be an open (available to the public) or closed (captive environment) app?
- 5 Does the app need to be a one-way communication tool or have the ability to communicate back and forth?
- 6 Do you need to capture data? If yes, what type and how much?
- 7 How will this app integrate, or not, with your website and other digital strategies and activities?
- 8 How will you communicate the availability and benefits of the app to your community?
- 9 Finally, once the app is deployed, how will it be distributed, and how will you promote the app?



QuickNote

Specs define the system's features, architecture and design. They can start out being a concept, and later evolve into a more detailed description of the app's functional characteristics.

QuickTerm

A **wireframe** is a two-dimensional illustration of a screen's interface that focuses on space allocation and prioritization of content, available functionalities and intended behaviors. They typically do not include any styling, color or graphics.

After having successfully led his team through the development process for numerous apps, systems and related projects, Ledoux shared the following best practices to help guide you along the way to mobile app creation:

- 1 Identify your minimum viable product (MVP).** This is the process of selecting the minimum required features that together, add the most value for the end user. For example, if your goal is to allow end users to adopt a pet, you may not have to include a chat feature with the pet's current owner in the first version of the app. An easier alternative would be to make the owner's contact information available. This lets you release your app quicker and gives you an opportunity to show your user base that you are coming out with new features. If this step is not performed, you may end up working for months – if not years – before releasing your app and validating your ideas.
- 2 Next, create high-level specifications.** Briefly describe the functionality. Don't go into too much detail at this time. These "specs" will help you define the system and provide a framework for more detailed requirements later on.
- 3 Create wireframes and prototypes** to help identify problems in the flow and usability early on, before spending top dollar on developing potentially broken features. User experience (UX) is the single most important factor you need to look at as early as possible to prevent usability disasters. You will want your app to be easy to navigate and intuitive.

- 4 Now start writing clear specs.** If possible, include some technical people (e.g., developers, information architects, quality assurance) who can help raise any feasibility issues early on. Detail every aspect of what the app should do. Also, spec in the right order, i.e., don't spec the stuff you won't do first.
 - Don't complete all specs before starting to code; that's what's called the "waterfall method" and it does not work well in software development. Some legacy organizations and older corporations generally tend to work in waterfall, which means they spend months or even years spec'ing a project, and by the time they develop the product – it's already out of date.
 - Specs should be done incrementally, almost in an artisanal fashion. This encourages and allows for discussion throughout the process, which can lead to the development of a feature that's closer to the overall vision, needs and reality than a spec that was written eight months ago.
- 5 Design high-fidelity mock-ups** – and don't forget to design popups, error messages and states, error pages, button states, etc.
- 6 Carefully choose between putting together your own development team or outsourcing.** Keep in mind that if you decide to outsource to another country, it may be difficult to manage if you are working across time zones. Also, standards in quality aren't always perceived the same way.
- 7 Choose a project management methodology** such as **Agile/Scrum** (recommended).

And for those who are more technically minded, Ledoux had some additional advice:

- 8 Choose your technology based on your needs.**
 - Will you need to scale?
Or do you need high availability, high security, multi-language, dynamic or static content, push notifications, etc.?
 - Native, web or hybrid?
iOS, Android or web? Swift, Java/Kotlin, JavaScript/HTML/CSS, etc.?
- 9 Think about how and who will maintain and support** the app after it's launched.
- 10 Plan who will deploy new versions and bug fixes.** And how? Also consider downtime, crisis management, etc.

QuickTerm

Agile prescribes working incrementally, collaboratively and flexibly. Agile commits the development team to two- to four-week "sprints" of work and provides them a method to deal with rapidly changing priorities and resource constraints. A few of the most popular frameworks that Agile teams adopt are Scrum, Kanban and Extreme Programming.



Apps change fast and can cost a lot to build and maintain

In the planning phase, think about how your app will evolve. If properly managed, updates can show users that the mobile app is regularly maintained and that your organization is invested in improving the mobile app over time.

Planning ahead for mobile app upgrades leads to reduced costs in the long term. Since updates are relatively easy to deploy, it becomes very easy to deliver new features progressively, as opposed to delivering them all at once. This approach is consistent with agile development methodologies, which allow you to focus on delivering the features that have the highest value first.

Apps often store data directly on the mobile device. As a result, updates can happen in several ways:

- The user downloads a new version of the app without the previous version installed.
- The user has a previous version of the app installed, has related data stored on the device and will install a newer version of the app. If the data storage format has changed in the new version of the app, a migration script within the app may be required to move from the old format to the new format. This must be taken into account when providing app updates to the public.

QuickAlert

Many factors influence the cost of native app development, including functionality, integration with existing databases and other unique requirements your organization may have. According to DigitalGov, MobileGov Community of Practice members report that development costs for Federal native apps range from \$50,000 to \$300,000.

In addition, maintenance can run from \$10,000 to \$12,000 per year for minor content and operating system updates.

static apps

Static apps don't need an internet connection to function and they don't change frequently. Static apps:

- Can be downloaded for the first time and all the content is built-in.
- Can be updated with an app update that contains the new or modified content.

This is an inefficient way to maintain an app's content as the lifecycle of updating, developing and deploying an update can take many months in some companies. However, it is a low-cost option compared to building a whole content management system (CMS) to push dynamic content.

dynamic apps

Dynamic apps need internet connectivity and change often. With dynamic apps:

- Content does not have to be distributed through an app update.
- Content is pushed to the apps, and the apps store the pulled content locally to offer offline capabilities.

This is an efficient way to update data. It is also the most complex, as it requires a CMS to manage and deliver the update (e.g., web portal, servers, push notifications) to different parts of the app.



It takes effort to **acquire and engage users**

Good content is key to audience engagement. That being said, you can prepare the content but how do you know anyone's reading it?

Acquiring and engaging users is a big challenge. You need to raise awareness to get users to go to the app store. Then you need them to download, install and open the app. And the next challenge is making the mobile app useful enough that they'll want to use it regularly.

To help measure your success, Apple and Google provide web-based tools that can be used to collect and analyze downloads and usage. In addition, analytics libraries can be built right into an app to find more granular details, such as the amount of time users spend on a specific page or the path they took to navigate from point A to point B.

Providing frequent updates, such as new high-quality content, is a great way to retain users. They will see that you are investing and constantly improving your app. Over time, this will entice users to come back and continue to use it. User retention is critical to the long-term success of any mobile app, making planning for regular updates critical.

QuickAlert

Push notifications allow you to connect with your audience for immediate communication – but don't send out too many. This is the number one reason why people uninstall an app.

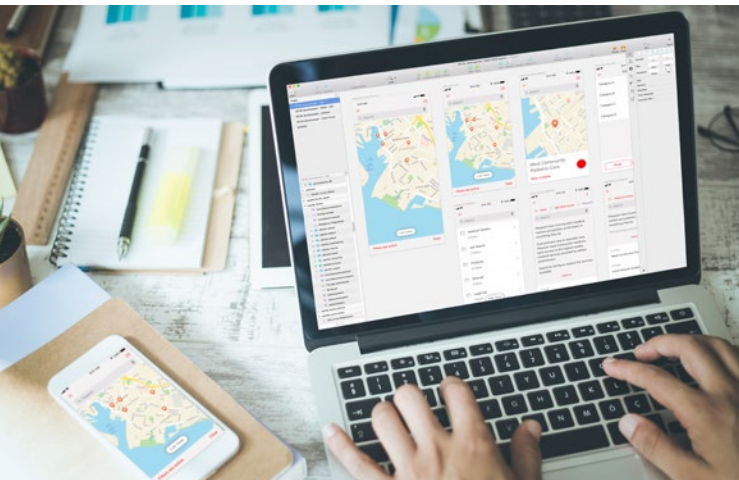


About QuickSeries®

QuickSeries® mobile app solutions are the ideal resource tools that are tailored to your community, facility or workplace. Our mobile innovation team includes developers, designers and content specialists who integrate vital and rich information into user-friendly systems. Our mobile app systems deliver the know-how families, residents and employees need for emergency and disaster preparedness, improving safety and security at home and at work, and achieving personal health and wellness.

Mobile Innovation Team

Using cutting-edge technology, our mobile innovation team can provide the most user-friendly app systems available on the market.



Modular Construction

Our systems are made up of modules, giving you the freedom to choose the features that will meet your needs and budget.

eGuide Library

The foundation of all of our apps is our QuickSeries® eGuides. Select the titles relevant to your audience from our vast eGuide library.



TOTAL SOLUTIONS FOR 360-degree outreach



 1-800-361-4653

 info@quickseries.com

 www.quickseries.com

You may not copy, distribute, transmit, email, reproduce, publish, post on a website, license, create derivative works from, transfer or sell this electronic file. © 2019 QuickSeries Publishing - September 6, 2019

