9-30-2019

Digital Health: Software as a Medical Device

Mauricio Novelo
Perelman School of Medicine, University of Pennsylvania

Nalaka Gooneratne
Perelman School of Medicine, University of Pennsylvania

James Weimer
University of Pennsylvania

Digital Health: Software as a Medical Device

Summary

- Software, such as mobile device apps or telemedicine, creates exciting new opportunities for patient engagement and for improving healthcare.

- There are three main types of mobile apps: native, web, and hybrid.

- The wireless technologies in smartphones and wearable sensors, such as smartwatches, offer the potential for additional biometric data collection.

- HIPAA compliance requires multiple levels of oversight and auditing.

- The software development costs for healthcare are considerably higher than for consumer-oriented products due to FDA regulatory requirements; testing out proof-of-concept through low-cost alternatives is an important development strategy.

Digital Health:
Software as a Medical Device

Mauricio Novelo,[1] Nalaka Gooneratne, MD, MS,[2] and James Weimer, PhD[3]

Idea | Exit

Topic Relevance by Timeline

Summary

- Software, such as mobile device apps or telemedicine, creates exciting new opportunities for patient engagement and for improving healthcare.
- There are three main types of mobile apps: native, web, and hybrid.
- The wireless technologies in smartphones and wearable sensors, such as smartwatches, offer the potential for additional biometric data collection.
- HIPAA compliance requires multiple levels of oversight and auditing.
- The software development costs for healthcare are considerably higher than for consumer-oriented products due to FDA regulatory requirements; testing out proof-of-concept through low-cost alternatives is an important development strategy.

Introduction

Over the past several decades, there has been an explosive growth in the use of software in patient care. This growth has been catalyzed by the development of enhanced hardware platforms that substantially augment processing speed, improved protocol standardizations allowing for increased security, growing patient and provider acceptance, and insurance/government initiatives that encourage the use of digital health software, such as the electronic medical record. As a result, academic entrepreneurs are increasingly interested in this space. However, working in digital health carries a set of unique challenges related to software development and validation, as well as Food and Drug Administration (FDA) oversight. These topics will be discussed in this chapter;

---

[1] Perelman School of Medicine, University of Pennsylvania
[2] Perelman School of Medicine, University of Pennsylvania
[3] University of Pennsylvania

while the primary focus is on smartphone app software, many of the core issues are relevant to other domains in digital health.

FDA Categories

The FDA broadly defines three categories of medical software (Center for Devices and Radiological Health):

1. Software that by itself functions as a stand-alone medical device: Software as a Medical Device (SaMD). This may also be described as a digital therapeutic, or a prescription digital therapeutic.
2. Software that is integral to the functioning of a physical medical device: Software in a Medical Device (SiMD).
3. Software used in the manufacture or continued maintenance of a physical medical device.

This chapter will be focusing on the first category, Software as a Medical Device. While Software as a Medical Device covers a broad range of potential applications, this type of software is increasingly used in mobile apps, such as smartphone software applications, that could be used by patients and caregivers for the diagnosis, management, and/or treatment of health and medical conditions.

The FDA exercises enforcement discretion on Software as a Medical Device that "pose minimal risk to patients and consumers" (Center for Devices and Radiological Health). These software functions include those that "help patients/users self-manage their disease or condition without providing specific treatment suggestions or automate simple tasks for health care providers" (Center for Devices and Radiological Health). The determination whether or not to regulate is made on a case-by-case basis and can vary according to situation and platform. It is also important to note that other regulatory agencies, such as the Federal Trade Commission (FTC), also play an important role, as demonstrated by recent FTC fines against smartphone app developers for deceptive advertising ("Lumosity to Pay $2 Million to Settle FTC Deceptive Advertising Charges for Its 'Brain Training' Program").

Potential Utility of Mobile Apps

For over half a century, significant efforts and resources have gone into improving healthcare through the development of software applications, such as telemedicine (Doarn et al.). Software that directly engages patients has the potential for increased levels of patient engagement, a model of participatory healthcare (Figure 1) that could improve patient outcomes while also lowering cost.

Figure 1. Participatory Healthcare Model.
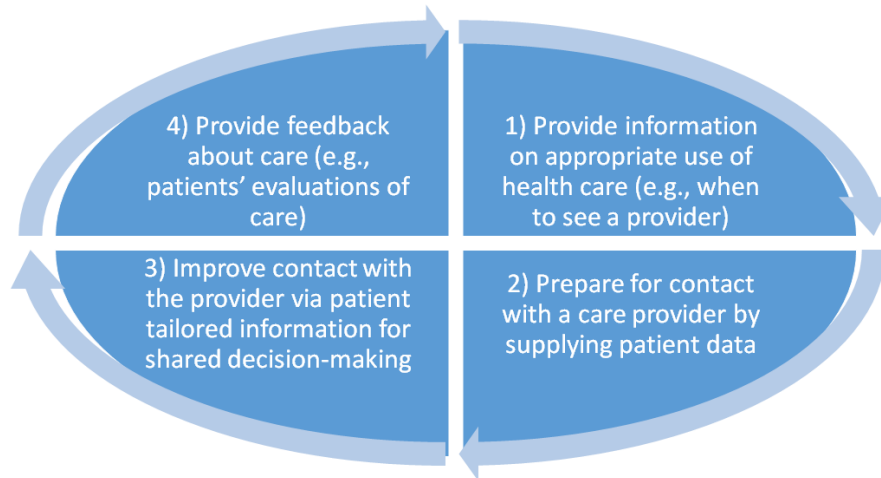
Source: Adapted from Wensing and Elwyn (Wensing and Elwyn)

In 2008, the iOS App Store completely transformed our relationship to software products through the creation of fully customizable, personal user experiences accessible from virtually anywhere in the world—i.e., the mobile app. Eleven years later, there are over five billion mobile users worldwide ("GSMA"). The ubiquity of mobile platforms in our everyday life has subsequently transformed mobile health (mHealth).

The uniqueness of mHealth lies in the transformative potential of the mobile app. Mobile devices are engineered for portability and deep personalization. In many ways, one's mobile device is more personal and private than their wallet, even as it simultaneously connects them to the wider world. Through active and passive data collection, mobile apps provide a means to continuously collect unique user data, process that data, and provide personal, data-based responses in near-real time. For clinical research, a mobile app could perform an ecological momentary assessment (EMA) at a randomized time or when the user meets some conditions, such as proximity to a specified GPS coordinate. For medical device trials, a mobile app's ability to leverage device components—including motion sensors, Wi-Fi, Bluetooth, and/or Near Field Communication (NFC)—provides a wide variety of ways to stay connected to a new medical device, running continuous analytics, data collection, or even big data processing on remote servers.

App Development Options

Through ongoing development and innovation, mobile apps have grown in complexity and design into three main categories: native apps, web apps, and hybrid apps. The main difference between these categories lies in their development. Ultimately, no one type of mobile app is better, faster, cheaper, or easier for all situations. However, for every research project hoping to include a mobile

app in any way, the choice of which mobile app type to use has drastic effects on timelines, budgets, enrollments, and outcomes. The following descriptions of each mobile app type include key areas to consider when choosing to use a mobile app in a project.

Native apps are typically written using a platform's "native" programming language, i.e., Swift for Apple iOS and Kotlin for Google Android. For native apps, Apple and Google grant the ability to access all device features and capabilities through their proprietary frameworks, development tools, and application program interface (API), written specifically for their respective languages. These benefits are effectively to entice loyalty from developers. Although similar in nature, Swift and Kotlin are not interchangeable, nor is any of their code currently reusable in other environments (e.g., on a server); therefore, for a mobile app to have an iOS and an Android version, it must be completely written twice. Many businesses choose to target users on a single platform to start, adding support for other platforms after gaining some traction. Such a strategy may not be appropriate for research because it would create an additional inclusion/exclusion criterion and potentially bias the data. For instance, iOS users have an average salary of $53,251 and spend an average of about five hours on their mobile devices, while Android users have an average salary of $37,040 and only spend an average of about four hours on their devices (Slickdeals). Nonetheless, a common denominator for both platforms is the dominance of each platform's proprietary app store. Having written an app using a particular platform's native programming language, the developer must then submit it into the platform-specific app store for distribution, pending review. This review is an evaluation of the app's functionality and user experience, and as such it is one of the most important parts of a native app. Because native apps are written in specific programming languages, using proprietary tools, frameworks, and APIs, the respective app store is essentially the only means by which a native app can exist. Failure to meet the app store requirements can lead to a native app being removed from the app store. The ongoing review process of every app consists of multiple review periods. These review periods take time and must be accounted for in project timelines. A review period typically starts upon submission, regardless of whether it is an initial submission or an app update. However, Google and Apple regularly perform additional reviews of each app listing, to ensure that these apps continue to meet their requirements. While the duration of the review period may vary, each review period effectively ends with a pass or a fail. It is not uncommon for an updated app submission to fail multiple times even if it had passed previously.

Research studies incorporating a mobile app can often take advantage of specific research and health-focused features a platform may offer, such as Apple's Healthkit; these features are constantly evolving, and thus a detailed discussion of them is beyond the scope of this chapter. However, there are some additional considerations that must be kept in mind that are relevant across platforms. Researchers must continue to attend to the maintenance of their mobile app, its user experience, and its status on the respective app stores to avoid interruptions in their availability/accessibility. Maintenance of a native app includes publishing updates that account for and

incorporate updates and changes to all review guidelines and underlying tools, frameworks, and APIs. Maintenance of the user experience is not as straightforward. Over the past decade, Apple has continued to enforce stricter guidelines while shortening the average review period to about 24 hours (Apple Inc, "App Review - App Store - Apple Developer"; *Saying Goodbye to App Review Times – Dave Verwer's Blog*). The Google Play Store has gradually enforced stricter guidelines while extending the expected review period up to seven days in most cases (Toombs; *Publish an App - Play Console Help*). These changes are symptomatic of the oversaturation of the mobile app market. From the first 500 apps launched in 2008, the number of apps available on the Apple App Store grew to over 2.2 million in 2017, and the Google Play Store peaked at 3.6 million in March 2018. Both figures have since declined to 1.96 and 2.46 million, respectively ("Topic: Mobile App Usage"; Liao; Ariel). Google and Apple intentionally seek to curate their app stores, updating their guidelines and requirements regularly. Section 4.2 of Apple's app development guidelines lists the minimum functionality of all apps, including those for research, on the Apple App Store:

> Your app should include features, content, and UI [user interface] that elevate it beyond a repackaged website. If your app is not particularly useful, unique, or "app-like," it doesn't belong on the App Store. If your App doesn't provide some sort of lasting entertainment value, it may not be accepted. Apps that are simply a song or movie should be submitted to the iTunes Store. Apps that are simply a book or game guide should be submitted to the Apple Books Store … Apps created from a commercialized template or app generation service will be rejected unless they are submitted directly by the provider of the app's content. These services should not submit apps on behalf of their clients and should offer tools that let their clients create customized, innovative apps that provide unique customer experiences. (Apple Inc, *App Store Review Guidelines - Apple Developer*)

In short, a mobile app must have a unique, compelling reason to specifically be a native app and not a web app. An example of such a compelling reason could be the ability to leverage Bluetooth or Near Field Communication to integrate a user's mobile device with some novel external sensor. Nonetheless, an app's reason will be reevaluated by Apple and Google as they continue to refine the requirements of a native app to be listed on their respective app stores. Unfortunately, their reasoning of which apps have unique, compelling reasons and which do not will remain subject to their discretion. Furthermore, this requirement disproportionately affects leaner operations such as small businesses, community organizations, and pilot research projects that do not have the means or funds to invest months of development into creating and maintaining a high-quality app with a unique and compelling feature set. For example, there already exists an abundance of mHealth-related native apps that simply administer a survey and/or present medically related information for educational and research purposes. Often, these apps are generated by some service or using some template. For instance, in November 2018 the Food and Drug Administration developed and released template code for creating these simple apps for Android and iOS ("New Real World

Evidence Digital Tool from FDA, MyStudies App"). The code is provided as is and does not guarantee passing the app store review. All of these apps must still offer a unique, compelling innovation beyond data presentation and collection to be considered fit for distribution on an app store. Essentially, an "app could be rejected just because the app store reviewer feels like there are already enough apps in [that] category" (Love, "How Is Apple Encouraging Progressive Web Apps?"). This tightening of criteria for native apps is an acknowledgment of the significant investment required to make a native app and an encouragement for developers to consider web apps for most simple use cases (Apple Inc, *App Updates for HTML5 Apps - News - Apple Developer*; Firtman, "Google Play Store Now Open for Progressive Web Apps").

Web apps are the result of recent innovations in internet browsers that allow web pages to be highly interactive and responsive. Currently, the progressive web app (PWA) is the most common standard for web apps. At its most basic level, any website can become a PWA by adding an app manifest and including a service worker. An app manifest is a single, "simple JSON [JavaScript Object Notation] file that tells the browser about your web application and how it should behave when 'installed' on the user's mobile device or desktop" ("The Web App Manifest | Web Fundamentals | Google Developers"). A service worker provides the mobile device with many of the necessary tools and information to essentially "run" a specific website more like a native app. While compliance with the progressive web app specifications varies across devices, a single progressive web app can currently operate on mobile devices (iOS and Android) and desktop computers (Firtman, "iPhone 11, iPadOS and iOS 13 for PWAs and Web Development"). Installation of a progressive web app is done directly from the website, without an app store or an app store review process. At this time, certain abilities are still experimental and/or not implemented for all devices, such as access to Bluetooth and push notifications on iOS. Their functionality offline is currently more limited than that of native apps. However, support is steadily growing, and Google now allows progressive web apps on the Google Play Store as Trusted Web Activities (Love, "Apple Safari Ships Service Worker & Progressive Web App Support"; Firtman, "Google Play Store Now Open for Progressive Web Apps"). A listing on the Google Play Store is particularly helpful because users expect all mobile apps to come from an app store due to the current minimal market recognition of progressive web apps.

Hybrid apps can be described as native app shells for web apps. They are an attempt to bridge the divide between native apps and web apps. At its most rudimentary implementation, a native app shell opens a browser window that displays the web app from either a website or from a JavaScript code that was bundled with the native app code. While hybrid apps generally have more access to the underlying mobile device than web apps, they are most commonly used as a means to distribute a web app via an app store since users have been trained to expect apps from app stores. There are multiple frameworks and services that will simultaneously generate the native app shell for both iOS and Android. Both versions would access the same web app. More advanced frameworks, such as React Native, create a bridge from the original JavaScript code to the native user interface

(UI) components of the mobile device, essentially rendering to the screen like a native app instead of like a web page. These advanced frameworks tend to require much more overhead, leading to apps that are much larger in size and take longer to load than a basic native app (Peal, "React Native at Airbnb: The Technology"). Moreover, for some use cases, such as supporting complex gestures, the amount of additional bridging code necessary to facilitate communication between the React Native code and the native code can lead to essentially supporting three platforms (React Native, React Native to iOS bridging code, and React Native to Android bridging code) instead of the single hybrid platform (Peal, "Sunsetting React Native"). The growing complexity of the bridging code led to Airbnb moving from React Native to writing purely native apps in 2018. This is not to say React Native has no valid use cases. Both Instagram and Facebook use React Native, for example. Often, the average user does not notice the difference between a hybrid app and a native app. At the time of this writing, there are many hybrid apps on both the Apple App Store and Google Play Store. Nevertheless, hybrid apps must still pass the app store review to be listed. Passing may become increasingly more difficult because Apple wants to remove all hybrid apps that solely open a browser window to a web app, which they refer to as HTML5 apps, from the Apple App Store by March 3, 2020 (Apple Inc, *App Updates for HTML5 Apps - News - Apple Developer*). The remainder of this section will address mobile apps in general and is applicable to native, web, and hybrid apps.

Opportunities for Data Integration with Biometric Sensors

Research studies using or developing biometric sensors benefit greatly from an integration with a mobile app. Bluetooth Low Energy (BLE) and Near Field Communication (NFC) are two wireless technologies, readily available in most mobile devices, that help biometric sensors overcome their limited computational bandwidth, storage space, and battery life. Using these technologies, a biometric sensor can quickly be "paired" with a mobile device. With an associated mobile app, data from the sensor can be streamed to the mobile device, which can store, process, and/or relay the data to a server for additional storage and processing. This process reduces the amount of local storage and computational bandwidth needed by a sensor. While a mobile app introduces an additional step in data retrieval from a sensor, the low energy consumption by BLE and NFC in comparison to Wi-Fi and cellular networks is often well worth the investment.

Institutional Considerations

Every mobile app should come with a user agreement that includes data use and privacy policies, especially those that handle sensitive information. The specifications within that user agreement can vary by institution. Each institution will have established software development best practices, in particular for Health Insurance Portability and Accountability Act (HIPAA) compliance. These best practices should be a part of the development process for a mobile app before any develop-

ment even begins. Generally, these will include recommendations and requirements for data transmission and data storage, assessments of security policies, delineation of maintenance and auditing responsibilities, and definition of acceptable use of internal and external resources. The recommendations and requirements for data transmission and data storage will generally mirror industry best practices for data encryption "at-rest" and "in-flight" and for multi-factor authentication. The assessments of security policies will cover how data breaches will be caught, mitigated, and reported. The delineation of maintenance and auditing responsibilities covers who will take responsibility of providing updates and bug fixes to the mobile app and its infrastructure (e.g., back-end servers) and who will track and review audit logs for suspicious activity, both malicious and bug related. Finally, the definition of acceptable use of internal and external resources establishes the kind of tools, services, and resources that a mobile app can use to mitigate a data breach and/or liability. For example, some institutions do not consider the use of Amazon Web Services servers as complying with HIPAA because of / resulting in the lack of a business associate agreement in place between the institution and Amazon (Office for Civil Rights (OCR)). This detail would be disastrous for a researcher to discover if their mobile app had already been built on Amazon Web Services. In this case, the researcher may need to substantially modify their data collection to limit it to de-identified data only, which will likely require discussion with the institution and the affiliated institutional review board (IRB).

Most academic healthcare institutions are also using an electronic medical record (EMR). Integration of an app with the EMR can substantially leverage the large body of data in the EMR and its ease of access for clinicians and patients. However, most EMRs are tightly controlled and access can be challenging and time-consuming. Several initiatives exist to enhance interoperability with the EMR and other health IT systems, such as the Fast Healthcare Interoperability Resource (FHIR) specifications developed by the Health Level Seven International non-profit organization (*SMART on FHIR*).

While digital health solutions often focus on the patient, it is also important to keep in mind the perspective of the healthcare provider when developing an effective software solution for deployment. Traditionally, healthcare providers have been reluctant to embrace digital health solutions for a variety of reasons (Figure 2); these must be addressed as part of the development cycle for an academic entrepreneur to successfully engage with this key stakeholder (see chapters on "Conducting Insightful Market Research" and "Human-Centered Design: Understanding Customers' Needs Through Discovery and Interviewing").

Figure 2. Factors Driving Low Levels of New Technology Implementation by Healthcare Providers and Institutions.
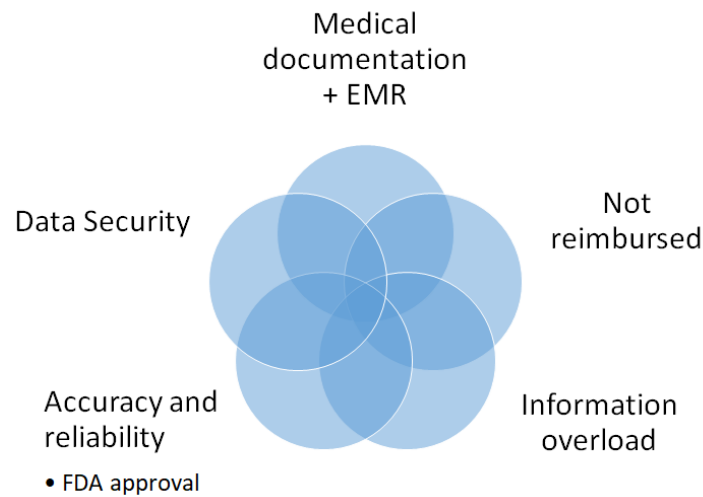

Intellectual Property

Digital health innovations often take a different approach to intellectual property (IP) as compared to drugs or devices. Software may be copyrighted, but there are often multiple approaches to performing a task, thus limiting the utility of a copyright to influence potential competitors from deciding to enter the space. In some cases, design patents can be used, but a competitor can easily find alternatives to those as well. In many cases, the startup team's deep knowledge of the clinical area and their ability to rapidly execute remain key ways to mitigate the lack of formal IP protections in digital health.

FDA Regulatory Oversight

The FDA has provided guidance regarding when digital health, such as mobile apps, may require FDA certification: "if a software function is intended for use in performing a medical device function (i.e. for diagnosis of disease or other conditions, or the cure, mitigation, treatment, or prevention of disease) it is a medical device, regardless of the platform on which it is run" (*Policy for Device Software Functions and Mobile Medical Applications—Guidance for Industry and Food and Drug Administration Staff*). FDA oversight is also required when medical claims are made in marketing material. The FDA has enforcement discretion, which means that, in some cases, the FDA may decide that review is not required. Broad categories of apps that may require FDA review are shown in Figure 3.

Figure 3. Mobile Apps That May Require FDA Review.
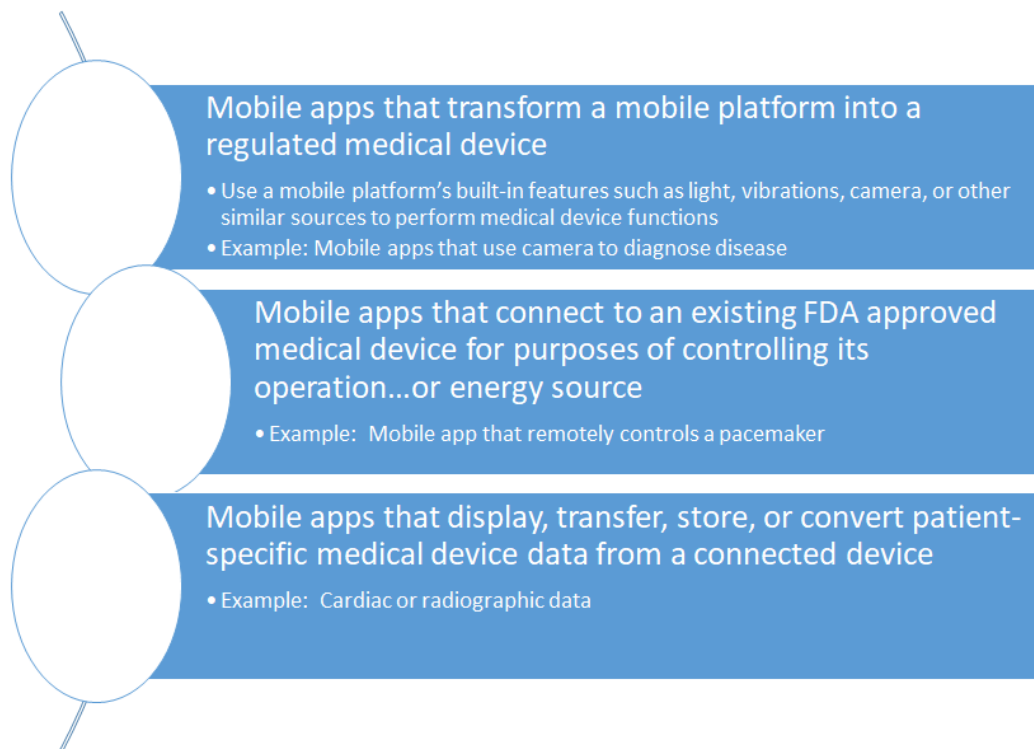
Traditional FDA certification pathways commonly used for mobile apps or digital health solutions require each product to be independently certified, such as through the 510(k) premarket notification pathway. In addition to the FDA template code for simple apps mentioned earlier, the FDA has introduced a new model (Pre-Cert) where the software company's development process is certified and will allow the company to create software devices without each new one undergoing its own FDA clearance/approval process. As described by FDA Commissioner Dr. Scott Gottlieb, it is intended for "certain novel types of low- to moderate-risk devices to obtain marketing authorization" ("FDA Looks to De Novo Pathway Model as It Unveils Updates to Pre-Cert Program").

For Pre-Cert, a software development company must undergo an excellence appraisal by the FDA first. This process includes evaluating quality system regulations (QRS) to demonstrate design control and validation. Dr. Gottlieb further clarified that the FDA "evaluat[es] the quality and excellence of the software developer for Pre-Cert. By collecting this information early, the Excellence Appraisal could be leveraged to streamline a developer's De Novo submission, reducing content the developer would need to submit to the agency under the De Novo pathway since the information would already have been demonstrated and documented during the Excellence Appraisal" ("FDA Looks to De Novo Pathway Model as It Unveils Updates to Pre-Cert Program").

Once a software development company has completed Pre-Cert, acceptable studies for SaMD approval are typically randomized trials. For example, Pear's reset-O software was approved based

on the results of a single unblinded randomized trial of 170 participants, which also rigorously monitored safety (adverse events) as well as efficacy ("FDA Clears First Prescription Digital Therapeutic for Opioid Use Disorder"). Additional guidance can be found in the "Software Precertification Program: Regulatory Framework for Conducting the Pilot Program within Current Authorities" document available at https://www.fda.gov/media/119724/download.

Cost

Developing a mobile app is similar to building a house. The cost will vary mainly by the level of customization, but any number of other factors might further shift the cost. If the desired feature set is simple enough, perhaps a cookie-cutter solution through an app generator might be best. However, as the feature set grows and becomes more original, the cost, timeline, and complexity will grow exponentially. This is because each feature will have to integrate to some degree with every other feature. Furthermore, features do not necessarily have to be additional components to the mobile app, but may also be on the back-end database or elsewhere; choices on the implementation of the back-end will affect more than just the cost, and thus merit additional discussion.

The back-end is the central database that stores all user data. There are multiple approaches for implementing a back-end, and not every approach fits all situations. To start, a back-end must have a location; either it is physically hosted in house/on premise or in the cloud. An on-premise back-end requires purchasing equipment, having a location to store and safeguard that equipment, and supporting the ongoing maintenance of the equipment. The physical proximity of the back-end to users will have a noticeable effect on data loading times. Meanwhile, cloud solutions have a variety of flavors, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and serverless. Infrastructure as a Service is the most comparable to in-house because it is effectively the contracting of a third party to host and manage all of the equipment that would be required for an in-house solution. A financial cost comparison of an in-house back-end versus one hosted with an Infrastructure as a Service is difficult to generalize for every situation. The internet has countless cost calculators, most from a cloud provider, all of which tend to disagree regarding the exact financial cost of any solution ("Total Cost of Ownership of Servers"; Deckler, "Cloud vs. On-Premises - Hard Dollar Costs Revisited"; Deckler, "Cloud vs. On-Premises – Hard Dollar Costs"). Cloud providers often cite how their solutions scale more readily than in-house solutions because cloud providers already have the equipment available at data centers around the world. Furthermore, their Platform as a Service and serverless offerings reduce the amount of configuration needed to maximize use of their Infrastructure as a Service. Platform as a Service includes managed tools and analytics—on top of Infrastructure as a Service—that can be helpful for knowing how many servers to provision and where to provision them (*What Is PaaS? Platform as a Service | Microsoft Azure*). Serverless solutions still use servers, but only when necessary. In-house, Infrastructure as a Service, and Platform as a Service all require servers to be running at all hours for an application to work. In contrast, serverless solutions provision the servers just in time when

they are needed by an application, in exchange for a small startup delay. Cloud providers only charge for the time a resource is in use.

Aside from variances in the cost of each approach, the configuration of the back-end has legal, security, and compliance ramifications, even more so than the type of mobile app. While cloud solutions make the argument that they are easier to sustain than in-house solutions, they undoubtedly give developers and administrators less control of the data than an in-house solution. If protected health information is to be sent to or stored in a back-end built on servers owned by a cloud provider, a formal business associate contract must be in place for compliance with HIPAA (Office for Civil Rights (OCR)); this typically increases costs. See the Resources section for a sample contract. Jeff Thomas, CTO of Forward Health Group, explained this in an interview:

> HIPAA compliance is always a dangerous and very vast term and healthcare organizations should always be leery of anyone selling a HIPAA compliant solution [...] Even if a solution enables you to use it in a compliant manner, doesn't necessarily mean it solves the compliance problem for you [...] Is the vendor you choose willing to sign a business associate agreement? If they hesitate or don't know what that is, they aren't the right vendor to choose because they don't understand your healthcare compliance needs when it comes to HIPAA. (HITInfrastructure)

Choices regarding the implementation of the back-end are just a tiny subset of all the choices that will affect the cost, timeline, and complexity of a mobile app. Unfortunately, there is the risk that cost estimates are thrown together based on optimistic guesses relying chiefly on anecdotal evidence; this can fundamentally undermine a credible business plan presented to potential investors, thus an academic entrepreneur is best served by carefully thinking through the options and obtaining multiple bids (see the chapter "Writing the Business Plan for a Life Science Startup or Clinical Program"). The remainder of this section will cover strategies that will generally help keep costs low and productivity high.

First and foremost, at least in the short run, there is no software product that is cheaper to build than to buy. Buying a software tool is the same as buying groceries from the local supermarket. Building a software tool is like buying a farm to grow your own produce. A common concern with buying an off-the-shelf solution is the loss of customizability. The options at the supermarket are limited, while a farm offers the opportunity to grow virtually anything. However, is the use case so unique and innovative that not a single product on the market can work, albeit less than ideally? Carefully consider the true worth of any gain a custom product might provide within the duration of the project. In terms of mobile apps, verify if a well-reviewed app or website generator might be sufficient, especially when a startup is in the early development phase and a basic pretotype using existing tools in a novel way would suffice (see the chapter "Rapid Prototyping Strategies"). For example, a website generator will put together an app almost instantaneously through a few clicks, no programming involved. For academic entrepreneurs, in some cases a basic app can be

developed in a computer science class, but this process can be challenging because the class will eventually end and the development team will disperse.

To determine whether a project can sustain a mobile app through gestation, consider the project's elasticity. If the project is solely to develop the mobile app, this calculation is easy: multiply by 1. If the mobile app is a secondary component of the overall project, a rough estimate would be to consider whether the primary component of the project could still be reached in half the time and with half the funds. This is not to say, spend the first half of the project on developing the mobile app and the second half on using the mobile app, because it incorrectly assumes that the mobile app will work perfectly during the second half of the project. Instead, create a critical path schedule with the primary non-app component and the mobile app development running roughly concurrently. Account for the app's continued maintenance. Bugs do not decrease linearly, but by some polynomial function. "The fundamental problem with program maintenance is that fixing a defect has a substantial (20–50 percent) chance of introducing another" (Brooks). Structure the mobile app's development as incremental growth. Start with a program that simply opens but displays nothing; then incrementally develop more functionality (Brooks). These increments must be very small, specific, and rapidly prototyped. User authentication is not a small or specific feature; instead consider creating a study form with pre-specified fields. If one has built a prototype with the desired feature, user tests will indicate more accurately the necessity of the feature, how long it would take to fully build the feature, its requirements, and its cost. Rampant in software engineering is the "assumption that one can specify a satisfactory system in advance, get bids for its construction, have it built, and install it" (Brooks). Instead, development of a mobile app should be seen as an iterative process that extends past the app's release. Costs will continue to accrue so long as the app exists, thus the app developer's history and cost overruns on past projects can be an important guide. Staying tightly engaged with the developers throughout each iteration would mean there will be fewer surprises regarding the costs or the timeline.

Conclusion

Digital health technologies offer tremendous potential for improving healthcare through their ability to engage patients and provide real-time data on medical conditions and treatment. Software can exist either as a medical device (Software as a Medical Device) or as part of an existing device (Software in a Medical Device). Mobile apps implemented through new smartphone devices and wearable sensors represent a growing field, especially due to their ubiquitous role in patients' lives. There are several important regulatory considerations guided by the FDA that can influence the potential development and use of a specific device. Thus, developing a new digital health solution may often require careful consideration of the feature set and prototyping to confirm the value/utility of specific features.

Resources

1. *The Mythical Man-Month* by Frederick Brooks is a timeless guide for how to approach software engineering, outlining common misconceptions of how software is built and how to avoid common pitfalls.
2. "HIPAA for Professionals": https://www.hhs.gov/hipaa/for-professionals/index.html. This is an invaluable resource for any application in healthcare.
3. Sample HIPAA contract, provided by the Department of Health and Human Services: https://www.hhs.gov/hipaa/for-professionals/covered-entities/sample-business-associate-agreement-provisions/index.html.

References

Apple Inc. "App Review—App Store—Apple Developer." *Apple Developer*, https://developer.apple.com/app-store/review/. Accessed 27 Sept. 2019.

App Store Review Guidelines—Apple Developer. https://developer.apple.com/app-store/review/guidelines/#minimum-functionality. Accessed 27 Sept. 2019.

App Updates for HTML5 Apps—News—Apple Developer. https://developer.apple.com/news/?id=09062019b. Accessed 27 Sept. 2019.

Ariel. *iOS Developers Ship 29% Fewer Apps In 2017, The First Ever Decline—And More Trends To Watch*. 30 Mar. 2018, https://blog.appfigures.com/ios-developers-ship-less-apps-for-first-time/.

Brooks, Frederick P., Jr. *The Mythical Man-Month*. Addison-Wesley Longman Publishing Co., Inc., 1995, anniversary ed., https://dl.acm.org/citation.cfm?id=207583.

Center for Devices and Radiological Health. "Software as a Medical Device (SaMD)." *U.S. Food and Drug Administration*, 2019, https://www.fda.gov/medical-devices/digital-health/software-medical-device-samd.

Center for Devices and Radiological Health. "Device Software Functions and Mobile Medical Applications." *U.S. Food and Drug Administration*, 2019, http://www.fda.gov/medical-devices/digital-health/device-software-functions-including-mobile-medical-applications.

Deckler, Greg. "Cloud vs. On-Premises – Hard Dollar Costs." *LinkedIn*, 8 Feb. 2016, https://www.linkedin.com/pulse/cloud-vs-on-premises-hard-dollar-costs-greg-deckler/.

"Cloud vs. On-Premises - Hard Dollar Costs Revisited." *LinkedIn*, 3 Feb. 2017, https://www.linkedin.com/pulse/cloud-vs-on-premises-hard-dollar-costs-revisited-greg/.

Doarn, Charles R., et al. "Federal Efforts to Define and Advance Telehealth—A Work in Progress." *Telemedicine Journal and E-Health: The Official Journal of the American Telemedicine Association*, vol. 20, no. 5, May 2014, pp. 409–18, doi:10.1089/tmj.2013.0336.

"FDA Clears First Prescription Digital Therapeutic for Opioid Use Disorder." *Medical Product*

Outsourcing, https://www.mpo-mag.com/contents/view_breaking-news/2018-12-11/fda-clears-first-prescription-digital-therapeutic-for-opioid-use-disorder/. Accessed 29 Sept. 2019.

"FDA Looks to De Novo Pathway Model as It Unveils Updates to Pre-Cert Program." *MobiHealthNews*, 8 Jan. 2019, https://www.mobihealthnews.com/content/fda-looks-de-novo-pathway-model-it-unveils-updates-pre-cert-program.

Firtman, Maximiliano. "Google Play Store Now Open for Progressive Web Apps." *Medium*, 31 Jan. 2019, https://medium.com/@firt/google-play-store-now-open-for-progressive-web-apps-ec6f3c6ff3cc.

"iPhone 11, iPadOS and iOS 13 for PWAs and Web Development." *Medium*, 20 Sept. 2019, https://medium.com/@firt/iphone-11-ipados-and-ios-13-for-pwas-and-web-development-5d5d9071cc49.

"GSMA." *GSMA*, https://www.gsma.com/. Accessed 26 Sept. 2019.

HITInfrastructure. "Cost, Flexibility Increase Virtualization's Health IT Appeal." *HITInfrastructure*, 29 Nov. 2016, https://hitinfrastructure.com/news/cost-flexibility-increase-virtualizations-health-it-appeal.

Liao, Shannon. "Apple's Total Number of Apps in the App Store Declined for the First Time Last Year." *The Verge*, 5 Apr. 2018, https://www.theverge.com/2018/4/5/17204074/apple-number-app-store-record-low-2017-developers-ios.

Love, Chris. "Apple Safari Ships Service Worker & Progressive Web App Support." *Love2Dev*, 8 Aug. 2019, https://love2dev.com/blog/apple-ships-service-workers/.

"How Is Apple Encouraging Progressive Web Apps?" *Love2Dev*, 10 Sept. 2019, https://love2dev.com/blog/apple-encouraging-progressive-web-apps-by-rejecting-apps/.

"Lumosity to Pay $2 Million to Settle FTC Deceptive Advertising Charges for Its 'Brain Training' Program." *Federal Trade Commission*, 4 Jan. 2016, https://www.ftc.gov/news-events/press-releases/2016/01/lumosity-pay-2-million-settle-ftc-deceptive-advertising-charges.

"New Real World Evidence Digital Tool from FDA, MyStudies App." *U.S. Food and Drug Administration*, 6 Nov. 2018, http://www.fda.gov/news-events/fda-brief/fda-brief-fda-launches-new-digital-tool-help-capture-real-world-data-patients-help-inform-regulatory-0.

Office for Civil Rights (OCR). "Business Associate Contracts." *HHS.gov*, U.S. Department of Health and Human Services, 16 June 2017, https://www.hhs.gov/hipaa/for-professionals/covered-entities/sample-business-associate-agreement-provisions/index.html.

Peal, Gabriel. "React Native at Airbnb: The Technology." *Medium*, Airbnb Engineering & Data Science, 19 June 2018, https://medium.com/airbnb-engineering/react-native-at-airbnb-the-technology-dafd0b43838.

"Sunsetting React Native." *Medium*, Airbnb Engineering & Data Science, 19 June 2018, https://medium.com/airbnb-engineering/sunsetting-react-native-1868ba28e30a.

Policy for Device Software Functions and Mobile Medical Applications—Guidance for Industry and Food and Drug Administration Staff. Food and Drug Administration (FDA), 27 Sept.

2019, https://www.fda.gov/media/80958/download.

Publish an App - Play Console Help. https://support.google.com/googleplay/android-developer/answer/6334282?hl=en. Accessed 27 Sept. 2019.

Saying Goodbye to App Review Times—Dave Verwer's Blog. https://daveverwer.com/blog/saying-goodbye-to-app-review-times/. Accessed 27 Sept. 2019.

Slickdeals. "iPhone Users Spend $101 Every Month on Tech Purchases, Nearly Double of Android Users, According to a Survey Conducted by Slickdeals." *PR Newswire*, 30 Oct. 2018, https://www.prnewswire.com/news-releases/iphone-users-spend-101-every-month-on-tech-purchases-nearly-double-of-android-users-according-to-a-survey-conducted-by-slickdeals-300739582.html.

SMART on FHIR. http://docs.smarthealthit.org/. Accessed 29 Sept. 2019.

"The Web App Manifest | Web Fundamentals | Google Developers." *Google Developers*, https://developers.google.com/web/fundamentals/web-app-manifest. Accessed 27 Sept. 2019.

Toombs, Cody. "[Update: Delays May Exceed 7 Days] Google Play Store Silently Extends App Review to All Submissions, Fails to Inform Developers." *Android Police*, 3 Sept. 2019, https://www.androidpolice.com/2019/09/03/google-play-expands-store-app-review/.

"Topic: Mobile App Usage." *www.statista.com*, https://www.statista.com/topics/1002/mobile-app-usage/. Accessed 27 Sept. 2019.

"Total Cost of Ownership of Servers." *SherWeb*, 21 Apr. 2019, https://www.sherweb.com/blog/cloud-server/total-cost-of-ownership-of-servers-iaas-vs-on-premise/.

Wensing, Michel, and Glyn Elwyn. "Methods for Incorporating Patients' Views in Health Care." *BMJ* , vol. 326, no. 7394, Apr. 2003, pp. 877–79, doi:10.1136/bmj.326.7394.877.

What Is PaaS? Platform as a Service | Microsoft Azure. https://azure.microsoft.com/en-us/overview/what-is-paas/. Accessed 29 Sept. 2019.

_____

Chapter Last Updated 9/30/2019.
Please check Scholarly Commons (https://repository.upenn.edu/ace/) for the most recent version.