TBM

CASE STUDY

Simplified Novel Application (SNApp) framework: a guide to developing and implementing second-generation mobile applications for behavioral health research

Jennifer Fillo, PhD,[1,4] B. Lynette Staplefoote-Boynton, MPH,[1,2] Angel Martinez, MS,[7] Lisa Sontag-Padilla, PhD,[1] William G. Shadel, PhD,[1] Steven C. Martino, PhD,[1] Claude M. Setodji, PhD,[1] Daniella Meeker, PhD,[3,6] Deborah Scharf, PhD[1,5]

[1] RAND Corporation, Pittsburgh, PA, USA
[2] Wake Forest Baptist Health, Winston-Salem, NC, USA
[3] Department of Preventive Medicine and Pediatrics, University of Southern California, Los Angeles, CA, USA
[4] Department of Psychology, University of Houston, Houston, TX, USA
[5] Simcoe County District School Board, Midhurst, ON, Canada
[6] RAND Corporation, Santa Monica, CA, USA
[7] RAND Corporation, Arlington, VA, USA

Correspondence to: J Fillo
jfillo@central.uh.edu

Cite this as: *TBM* 2016;6:587–595
doi: 10.1007/s13142-015-0363-7

Abstract

Advances in mobile technology and mobile applications (apps) have opened up an exciting new frontier for behavioral health researchers, with a "second generation" of apps allowing for the simultaneous collection of multiple streams of data in real time. With this comes a host of technical decisions and ethical considerations unique to this evolving approach to research. Drawing on our experience developing a second-generation app for the simultaneous collection of text message, voice, and self-report data, we provide a framework for researchers interested in developing and using second-generation mobile apps to study health behaviors. Our Simplified Novel Application (SNApp) framework breaks the app development process into four phases: (1) information and resource gathering, (2) software and hardware decisions, (3) software development and testing, and (4) study start-up and implementation. At each phase, we address common challenges and ethical issues and make suggestions for effective and efficient app development. Our goal is to help researchers effectively balance priorities related to the function of the app with the realities of app development, human subjects issues, and project resource constraints.

Keywords

Mobile applications, Software development, Best practices, Mhealth, Methodology, Health

Implications

Practice: Practitioners can apply methodological issues described here (e.g., design, testing, data protections) to self-monitoring and biofeedback interventions administered via smartphone.

Policy: When evaluating research proposals, institutional review boards need to consider whether researchers have adequately addressed the security issues unique to the handling of human subjects data collected in participants' natural environments and saved and/or transmitted over smartphone devices.

Research: Behavioral health researchers should consider using the SNApp framework to help structure and simplify the process of developing and implementing second-generation (e.g., multimodal, increasingly complex) mobile applications for health-related research.

Advances in mobile technology have opened up exciting possibilities for researchers studying health and health behavior. What started as a means to improve the quality of self-report data collected in the field (i.e., through electronic time stamps) [1] has exploded into numerous options for researchers seeking to combine information from, for example, self-reported subjective experience, objective measures of health information (e.g., actigraphy, blood pressure), and other factors such as location and environment (e.g., geospatial monitoring, air quality). However, behavioral scientists aiming to develop second-generation (i.e., multimodal, increasingly complex) mobile applications (or "apps") will encounter issues ranging from how best to structure an app development study, to testing the app itself, and a host of technical and ethical considerations related to this emerging data collection approach.

The purpose of this paper is to present our Simplified Novel Application (SNApp) framework (see Fig. 1) as a practical guide for researchers interested in developing and using second-generation mobile apps for collecting data in health-related research contexts. The intended audience is researchers with knowledge and expertise in behavioral health science but limited expertise in mobile technology and app development, since expertise in both is necessary for the success of such projects. Unlike earlier guides that addressed more foundational issues associated with simpler app-related data collection (i.e., prepackaged, single function), our focus is specifically on smartphone-based apps that can be combined to collect health-related data through multiple channels and

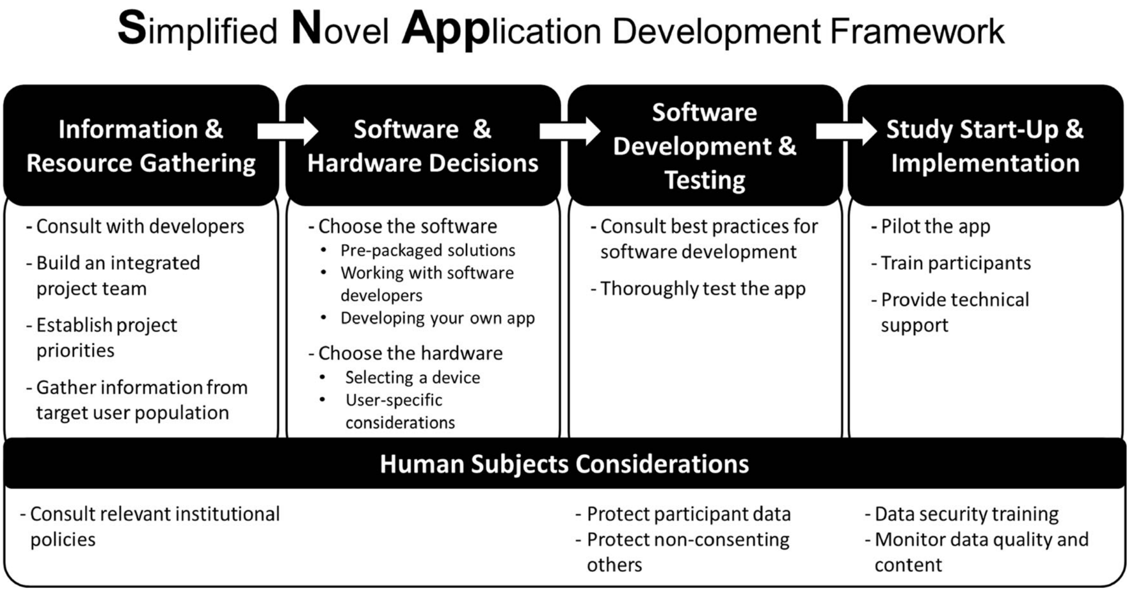
Simplified Novel Application Development Framework
Information & Resource Gathering
Software & Hardware Decisions
Software Development & Testing
Study Start-Up & Implementation
- Consult with developers
- Build an integrated project team
- Establish project priorities
- Gather information from target user population
- Choose the software
• Pre-packaged solutions
• Working with software developers
• Developing your own app
- Choose the hardware
• Selecting a device
• User-specific considerations
- Consult best practices for software development
- Thoroughly test the app
- Pilot the app
- Train participants
- Provide technical support
Human Subjects Considerations
- Consult relevant institutional policies
- Protect participant data
- Protect non-consenting others
- Data security training
- Monitor data quality and content

Fig. 1 | The Simplified Novel Application (SNApp) development framework

the project management, institutional, and human subjects issues associated with this research. Guides for software developers abound, but we know of no other framework that specifically addresses behavioral health researchers' needs in the app development process. The SNApp framework fills this gap. In this paper, we illustrate SNApp's utility with lessons learned from a case study of developing a second-generation mobile app that simultaneously collects text message, voice, and self-report data.

CASE STUDY OVERVIEW—A MULTIMODAL, IN VIVO ASSESSMENT OF ADOLESCENT SEXUAL COMMUNICATIONS WITH PEERS

Peers are an important influence on adolescent sexual behavior. For instance, if friends hold favorable attitudes towards having sex, then youth report stronger intentions to have sex, higher rates of having sex, and shorter times to sexual initiation [2–4]. Yet, very little is known about *how* adolescents get information about sex from their peers. With a grant from the National Institute of Child Health and Human Development (NICHD), we developed an app that captured voice and text messages, as well as self-report data, to investigate the context of teens' communications in their natural environments. The text message sampling and collection of self-report data using ecological momentary assessment (data from brief, self-report surveys completed at random intervals, multiple times per day in participants' natural environments) [1, 5] were easily built from existing software components. However, new software development was necessary to capture random snippets of participants' speech (reported previously) [6] while filtering out non-participant speech. We outline the SNApp framework, drawing upon our experiences developing this app.

SNApp FRAMEWORK

Our SNApp framework divides the development and implementation of a second-generation behavioral health mobile app into four phases: (1) information and resource gathering, (2) software and hardware decisions, (3) software development and testing, and (4) study start-up and implementation.

Phase 1: information and resource gathering

Steps in this phase include (a) consulting with developers, (c) building an integrated project team, (c) establishing project priorities, (d) gathering information from the target user population, and (e) consulting relevant institutional policies that may affect the project.

Consult with developers

We recommend that behavioral scientists consult with an app developer early in the project planning process; ideal developers will have expertise relevant to the functional priorities for the app. Developers may help identify an existing app that can be used or adapted and identify components that must be built anew. An experienced developer may also advise on the skills required for the development team. For example, if the proposed app is dependent on external database calls, or if graphic design is critical, these skills can be identified and appropriate staffing can be done in advance. In addition, developers can help rank proposed app features by degree of development difficulty and associated risk. With this information, researchers can make strategic decisions about whether or not all proposed app functions should be included in the final product. In some university settings, these kinds of development challenges may be "adopted" by computer science classes or teams; this kind of partnership may be a cost-effective solution for smaller budget projects.

Build an integrated project team

We strongly recommend that the primary app developer (s) be integrated into in the core project team, attending meetings and contributing to all stages of the project. This will help ensure that the project leader allocates adequate time for app development and testing. For instance, programmers may be able to foresee additional time needed to pilot aspects of the software (e.g., user interface) that other team members may not anticipate. To ensure that the app supports the investment made in the rest of the project, it is imperative that programmers and the software development team get involved early and stay involved throughout the project.

Establish project priorities

Next, the project team should work together to establish clear priorities for the overall project. Whereas project priorities may shift during the development process, adding functionality to an app that is already under development can be complicated; having a clear picture of the app's essential functions from the outset will help to avoid labor-intensive and expensive revisions later on. Other questions investigators should consider are whether the app is intended to be maintained, extended, and updated beyond the life of the current project. More generalized application architectures help to minimize the impact of future device and operating system changes and facilitate the addition of new functionality. However, more generalized application architectures are often more difficult to put in place. At this point in the development process, researchers should also carefully consider whether simpler study designs might suffice.

Gather information from the target user population

Early information gathering from the target user population can help researchers target various parameters of the app and test the acceptability and feasibility of its design. This can help researchers avoid investing in app features that will not get used in the field [7, 8]. User feedback can be gathered from potential participants and others who will affect (e.g., parents of minors) or be affected by (e.g., spouses) participants' use of the app. In our study, we primarily relied on focus groups to collect this information [9, 10], but researchers can also conduct structured or semi-structured interviews, surveys, observations of mobile device or app use, or diary studies, among others [11]. Below are some app-related issues that arose during our initial information gathering.

Comfort with mobile devices and the app–Although smartphone ownership is widespread and growing [12, 13], researchers should not assume technological proficiency among potential participants. For example, older and lower income populations may have less experience using mobile apps [14]; populations with low-level literacy/numeracy may struggle to make use of data "pushes" including quantitative information or text-based instructions.

Willingness to have app installed on phones–It is important to consider whether participants' phones will support the app and the security requirements of your institutional review board (IRB). Our focus groups revealed that many teens did not have devices that could support our study app, so we opted to issue participants a research device instead (discussed further below). When considering using study-issued devices, it is important to investigate participants' willingness to forgo their own devices during the study and endure the temporary inconveniences of doing so (e.g., limited features on a new device, difficulty transferring contacts, using a new number). In some cases, it may be possible for participants to retain their own phones for personal purposes in addition to the research device, but this was not possible in our study because we needed participants to make all of their calls and texts on a single device.

Importance of device features or accessories–Different research populations may prefer different device features and accessories. We asked pilot participants to consider several devices and microphones for recording their speech. Unexpectedly, we learned that teens thought Bluetooth earpieces were "for old people," which guided our decision to use an earbud-style headset/microphone. Additionally, some teens were unwilling to participate in the study if they had to forgo the music libraries stored on their phones. We opted to proceed without participants unwilling to make this concession.

Obstacles to protocol adherence–It is important to understand how potential participants will use the devices and potential obstacles to adhering to study protocol (e.g., work-related restrictions). This information can inform hardware and accessory selection, app development, and the development of relevant training materials. In our study, participants noted that particular cell phone "skins" (i.e., soft, protective cases) increased the appeal of the device and their willingness to use it in public. Male participants (in particular) noted that these cases would increase their ability to keep the phones with them at all times.

Human subjects considerations

Consult relevant institutional policies–Projects involving mobile devices may raise issues regarding the storage, transfer, and disposal of data that are different from those in which data are stored in secure laboratory spaces; these devices are constantly moving, may be Web connected, and are potentially less under the researcher's control. Mobile device data protection requirements may be reduced if the devices immediately transmit (and do not store) encrypted data to secure servers. In cases where data does need to be maintained on the study device, or when researchers are accessing study data in real time, extra protections may be necessary. In our study, the device we had considered early in the app development process did not have the encryption capabilities required by our

Table 1 | Pros and cons of three major software solutions for native mobile application development

	Prepackaged software solutions	Working with a software developer	Developing your own application
Pros	Ease of use Saves time: -Software skeleton is already developed -Will require less testing time Less expensive than hiring a software developer	Greater flexibility to design app to meet your exact needs Software developers can serve as a valuable resource during the entire app development and study design process	Greater flexibility to design app to meet your exact needs Can save money depending on complexity of app and your own expertise level
Cons	Limited to capabilities of existing software package Changes to software (if possible) come at an additional cost	Most expensive option Testing process for building application from scratch is very time consuming	Can be very costly in terms of time required to complete Testing process for building application from scratch is very time consuming Do not have software developers as a resource for the project

institution for sensitive data, so we had to switch devices before completing software development.

Phase 2: software and hardware decisions

There are innumerable software and hardware options for mobile app development, and they vary in complexity, flexibility, and price. Whereas we address both software and hardware considerations in turn, neither set of decisions is secondary to the other. Further, hardware and software decisions are interdependent, as applications developed on the Android and iOS platforms can only be run on compatible devices. In our experience, careful consideration, documentation, and reporting of choice points in phases 1 and 2 will likely strengthen the quality and acceptability of competitive funding applications undergoing peer review.

Choose the software

Researchers may obtain prepackaged software, work with a software developer, or develop an app without the assistance of a developer. Table 1 details the benefits and drawbacks of each approach. Because a single-software approach may not work for all projects, we recommend that researchers walk through these options separately for each of the core functionalities of the app. For example, software was easily modifiable for our participant self-report surveys (EMA) and text message data. However, we needed to develop our sound-recording software virtually de novo, working closely with highly experienced programmers so that we did not record voices from non-consenting others (per IRB requirements and state law).

Prepackaged software solutions–Many companies offer prepackaged software solutions that include a skeleton for the app and allow researchers to input their desired content (e.g., survey question content) and assessment schedule [15], reducing (but not eliminating) time spent testing the package in the field. For example, open-source apps created using the recently released Apple ResearchKit [16] may meet needs of research projects that do not require support for Android.[1] On the whole, these options may be cheaper than building an app from scratch. However, prepackaged options may not meet all of a project's needs. Companies may be willing to make changes to meet user-specific needs, but these may still have certain limitations and/or come at an additional cost. In our study, existing EMA software could largely handle all of the types of questions we wished to ask, but programs were unable to issue surveys according to the event-related schedule that we required. For second-generation apps, programming expertise may still be needed if researchers need data or devices to interact.

Working with software developers–A software developer or team of developers can build an app precisely suited to researchers' needs as well as integrate data from multiple instruments or channels that have not been combined before. Developers can include computer science colleagues and their students (likely at lower cost but longer timeline) or private software developers. At some universities or research institutes, a team of developers might be part of the organization and thus be familiar with researcher needs. Some software development firms cater to researchers; however, others may not be familiar with the research process. The ability for software developers to anticipate IRB-related data protections and other research-specific needs should be taken into account when selecting a software development partner. Developers can become valuable resources throughout the research process, providing helpful suggestions about

[1] At the time of publication, open-source apps were available to collect data related to a variety of conditions, such as diabetes, asthma, Parkinson's disease, cardiovascular disease, and breast cancer.

navigating hardware-related decisions, what to focus on during testing and piloting, and how to design user training. A novel final product may also be sharable (or marketable) to others conducting similar kinds of research.

Ultimately, the decision to develop an app "from scratch" must be weighed carefully against the project's budget and time constraints. Hiring software developers to build, test, and refine a custom mobile app can be expensive (note that developer time may cost more than senior behavioral scientist time) and time consuming [17] (although potentially time saving in the long run if the final product works well). These processes should not be rushed, as rushing could compromise the quality of the final product.

Developing your own app–Researchers may also choose to build their mobile app without the assistance of a professional developer. The feasibility of this route depends on the complexity of the project and the researchers' own technological knowledge. Currently, there are two dominant platforms for mobile app development: iOS (Apple) and Android (Google), although other competitive options (e.g., Windows, Amazon) are emerging. Both Apple and Google provide materials to assist with app development.[2] Third-party resources for new developers are also available.[3] Briefly, our experience suggests that the Android platform allows for greater development flexibility than does the iOS platform.[4] Google's more open-source perspective can be friendlier to researchers and their needs. Apple tends to have more restrictions on how their devices and software can be used.[5] However, software platform decisions need to be made in conjunction with hardware decisions, given that the device (s) researchers chose may limit the software that can be used. For our study, colleagues provided us with iOS code for sampling of voice data. However, the filtering of non-participant speech would have required significant modification for this critical app function; this precluded us from using the Apple product. In short, while developing your own app may appear like the most affordable option, it may ultimately be more costly in terms of time, quality, and flexibility of the final product.

Choose the hardware

Key functionalities will likely drive researcher decisions about hardware. These may include, for example, the ability to interface with other devices (e.g., an accelerometer), the capability to upload or download data via Wi-Fi or cellular networks, or necessary storage capacity. More so than software decisions, hardware decisions are also likely to be heavily influenced by characteristics of the intended users (i.e., their priorities, preferences, comfort with technology) and the contexts in which the devices will be used.

Selecting a device–The primary hardware-related decision is choosing the device on which the app will run (see Table 2).

Many researchers opt for using *participants' own devices.* This can be cost-effective, because investigators do not have to purchase hardware and phone plans. It may also improve compliance because participants are likely to regularly carry their own devices. However, many populations may not have devices that support the study app, meet data security requirements, or interface with any additional devices. Additionally, apps that require newer, more expensive devices may skew the study population towards participants who can afford and choose to purchase them [14, 18].

Issuing a research device eliminates sampling bias concerns and affords researchers much more control over app functioning (including providing more targeted technical support to participants in the field) and greater data security than when using participants' devices. However, there are significant logistical challenges associated with issuing research devices and cellular data plans (see "Gather information from the target user population"). Further, research devices may not need to be functioning phones. This avoids potential interference between the app and the phone function, and it may be less expensive. Although much of the early EMA research was completed successfully with non-phone (i.e., electronic diary or palm-top computer) devices [19, 20], these devices are less common and may result in lower compliance, because participants are not inherently motivated to carry a secondary device.

User-specific considerations–Researchers should consider the context in which the device and app will be used, given they may have implications for recruitment, protocol compliance, and data quality. If users need to carry the device with them at all times, smaller devices may be preferable. Depending on where they will be used, protective cases may be important. Whether devices need to remain connected to the Internet (e.g., to access remotely stored information, to transfer data to a remote secure server) can significantly affect overall cost and may limit the service providers that can be used (e.g., because of coverage quality, plan costs), as well as the battery life of the device.

[2] Health researchers may be particularly interested in the Apple ResearchKit: www.apple.com/researchkit. It is an open-source framework that aids in the creation of apps for medical research by utilizing the sensors and processing power built in to iPhones.

[3] We have found the Massachusetts Institute of Technology resources (e.g., tutorials, forums) for developing Android applications to be particularly helpful: appinventor.mit.edu.

[4] Apple is more restrictive about what developers are allowed to alter about the phone and its standard functions (e.g., text message interception, low-level audio access). Developers have more flexibility to alter device functions to meet software needs with Android.

[5] Whereas both charge annual fees to develop and sell applications through their platforms (Apple $99, Google $25), Google allows non-market apps to be installed on Android devices. Apple's "App Store" requires approval before an app can run on an iOS device, but an iOS developer can provision up to 100 devices (owned by the developer) per year for testing purposes without approval.

Table 2 | Pros and cons of using participants' devices vs. research devices to run your native mobile application

	Participants' devices	Research devices	
		Replace participants' phones	Separate device
Pros	Significantly cheaper for researchers compared to research device Convenient for participants Likely to increase compliance	More control for researchers than using participants' phones More control over data security than using participants' phones Easier to provide technical support Fewer population limitations than using participants' phones Single-device benefit	Avoids potential interference with phone function Can be less expensive than phone options
Cons	Other apps or functions of the device can interfere with app functionality Harder to anticipate problems Less control over data security Limited to populations with requisite devices	Participants must be willing to give up phone Added expense of calling and data plans Problems associated with transferring contacts and forwarding calls	Likely to have lower compliance than with functional phones Easier for participants to forget than device that also functions as their phone Very few device options currently on the market

Participants' day-to-day lives may also influence their ability to use the device and app (and vice versa). For example, we limited data collection to the summer months because teens are usually prohibited from using phones during school. Some participants required hardware that could be worn discretely (e.g., under work clothes) or that issued silent alerts (e.g., vibrating). Similar considerations for accessories or accompanying data collection devices should be made at this stage in the project.

Phase 3: software development and testing

Steps in this phase include (a) consulting best practices for software development, (b) thoroughly testing the app, and (c) taking steps to protect participant data and data that may be collected from non-consenting others.

Consult best practices for software development

Like clinical practice guidelines, software development best practices may help researchers and programmers create the best product possible within the project scope and with the resources available. Many new programmer-oriented software development guides and resources are now available. We also recommend consulting resources that describe best practices for effective user-centered, iterative design [11, 21], including a classic work by Larman and Basili [22].

Given that how participants engage with mobile apps is not as well understood as other data collection tools (e.g., telephone or survey instruments), we recommend that readers consider variants of AGILE development practices for their behavioral science projects [23, 24]. AGILE is typically compared to a "waterfall" approach [25, 26], which begins with a well-known set of dependencies and requirements based on well-defined models and "blueprints" (as would be the case for very straightforward data collection apps that rely on hardened and well-tested aspects of the mobile platform and do not involve introducing novel or interactive features). In contrast, AGILE practices are based on iterative and incremental development cycles; app requirements and solutions evolve through collaborations among all key members of the project team (e.g., psychologist, physician, statistician, programmer). AGILE methods have implications for the project timeline since testing of the app occurs through rapid cycles of stepwise changes to the app, instead of, for example, a single-project piloting phase, as is common in many laboratory study designs. Ultimately, each of these approaches has costs and benefits [27–30], and the best solution will depend on the nature and complexity of the development project.

Thoroughly test the app

Constant iteration and course checking during software development is key, and resources need to be saved for addressing issues that arise once the app is used in the field. Resources for this testing process should be allocated proportionately to the complexity of the app. The testing process should ensure that the app works as intended, investigating different ways users may interact with the app and the device, as well as seeking out the conditions under which it will fail. Testing should include different types of individuals (including the project team, colleagues less familiar with the project, and individuals from the intended participant population) in order to reveal issues that may not be revealed by a single type of user. Additionally, testing should be conducted systematically and iteratively; problems are discovered and fixed, and the app is tested again, because fixing one problem may cause new problems. Failing to complete thorough testing, although time consuming, can cause considerable delays in the study, lead to the collection of unusable data, and overall cause significant problems for researchers. Detailed testing records are critical for providing technical support to participants later on in the field (discussed more below).

For apps to be used beyond single short-term research projects, researchers should also plan for required maintenance of an app. Updates to the software may be required to ensure that the app remains functional, relevant, and useful as new advances in operating systems and other device features emerge. For instance, updates to old operating systems may interfere with the function of an app designed to work well with an earlier iteration.

Human subjects considerations

Protect participant data–Researchers should use multiple layers of security to protect participant data. This can include password protections on the user-directed interfaces for the mobile device and study app (to preclude data entry from non-participants), as well as on the researcher-directed interfaces. This may include further passwords, as well as encrypting the data collected on the device (so that if someone were to access the data, they cannot not discern its content). In some cases, it may also be advisable to limit participants' access to certain functions of the device. We disabled the camera and Internet capabilities of the phone in our study to prevent users from downloading anything that might compromise the security of their data (e.g., software virus) and to protect researchers from ending up with information (e.g., "sexts") that could put participants at risk. However, researchers should carefully consider whether this additional step is necessary given their study population and focus; limiting device functionality may interfere with protocol adherence or serve as a disincentive to study participation altogether. Very few of our potential participants refused participation for this reason.

Protect non-consenting others–In studies involving data collection via mobile devices, information may be inadvertently gathered from individuals who have not consented to participate (a.k.a. collateral participants). In our study, we took several steps to reduce the risk of collecting data from collateral participants including password-protecting the device, using software to filter out collateral participants' speech, and explicitly training users in the safe and responsible use of the device and app (including not sharing the device with others). Further, we gave participants the opportunity to delete any texts or sound files that might have contained information about non-consenting others. Despite these protections, the possibility of collecting some collateral participant information remains. Researchers should work closely with their IRBs to have a plan in place before the start of the study to ensure a rapid and effective team response in the event they collect data from collateral participants.

Phase 4: study start-up and implementation

Steps at this phase include (a) piloting the app, (b) training participants, and (c) providing technical support. We also address ethical considerations relevant to participant training in data security and monitoring data quality and content.

Pilot the app

Piloting the study with a fully functioning app serves several important functions. In addition to helping researchers practice the protocol, work out problems in methodology, and verify that procedures work properly, piloting aids in testing the final app and refining training and technical assistance materials. Our piloting revealed that a handful of teens were capable of, and interested in, circumventing a parental control app that we use to preclude access to the phone's camera. Consequently, we took further steps to disable the camera before data collection began.

Train participants

Thoroughly training all participants (including "tech savvy" individuals) in the use of the app is critical, as the quality of your data depends on all participants interacting with the device in the same way. This includes explaining what participants should use the device for (e.g., making calls, answering surveys, logging events of interest), what they should not use it for (e.g., storing personal photos, downloading recreational apps), and steps to ensure that the device functions as intended in the field (e.g., how to wear, charge, and protect equipment; signs the device/app is malfunctioning). We implemented a brief multiple-choice test during the initial training session (e.g., with scenarios about a malfunctioning device) to ensure that participants understood key information about device function, care, and data safety. We also had participants engage with the device as they would in the field to reveal any remaining issues with function or compliance before they leave. In our study, the quality of our voice recording data improved once we implemented explicit training and demonstrations of how and where (on the lapel) to wear the microphone. Researchers should anticipate that the development of training materials/procedures will be iterative, based on feedback from participants and trends in the type of problems reported to technical support personnel.

Provide technical support

Researchers should budget for technical support to participants while they are using the app in the field. Technical support personnel (e.g., well-trained research assistants or graduate students) can serve several functions including logging device malfunctions (to support later data cleaning) and compiling a list of software and hardware problems and solutions. Fortunately, many problems and solutions can be anticipated based on problems identified during testing. Effective technical support can improve project data completeness (minimizing periods of device malfunction), data quality (ensuring that participants are interacting with the device, as intended), and participant satisfaction (reducing unanticipated visits to the study site for help). It can also reduce study costs and time associated with unanticipated supplementary data collection.

Human subjects considerations

Researchers should routinely monitor the quality and content of data collected to identify app and device malfunction and subterfuge (from participants or others). Given that mobile devices may capture sensitive, private information from participants in their natural settings, efforts to maintain the integrity of app data should be ongoing. For protocols including study-issued devices, researchers may offer participants the opportunity to expunge any personal information (e.g., photos, text messages) from the device before turning it back to the study staff for data collection and/or device maintenance.

CONCLUSION

Advances in mobile technology and mobile apps have opened up an exciting new frontier for researchers interested in studying health and health behavior. This second generation of mobile apps has enabled, for the first time, the simultaneous collection of multiple types of data in real time. Sophisticated mobile apps allow these processes to easily happen in real-world contexts, minimizing disruption to users' everyday lives. However, with these advancements comes a host of decisions and challenges that researchers must address.

We have proposed the SNApp framework, which can help behavioral health researchers to structure and simplify the process of developing and implementing studies using second-generation mobile apps for data collection. We have described many unique challenges and numerous decision points associated with second-generation app research, as well as practical suggestions for researchers new to these methods. The possibilities for what researchers can accomplish with mobile apps will continue to grow with rapid advancements in mobile technology. When designed and executed effectively, app-based research has the potential to contribute to significant advances in behavioral health research.

Acknowledgments: This work was supported by grant R21 HD067546-01A1 from the National Institute of Child Health and Human Development awarded to Deborah Scharf, Steven Martino, William Shadel, and Claude Setodji. The authors would like to thank Sarah Hauer, Stacey Gallaway, and Robert Hickam for their administrative support with the grant. They would also like to thank Matthias Mehl for his advice on how to use smartphones to capture speech.

Compliance with ethical standards

Conflict of interest: The authors declare that they have no competing interests.

Adherence to ethical principles: All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional research committee. All authors completed human subjects training prior to conducting the research, and all authors maintained up-to-date training throughout the course of the project. All participants gave informed consent prior to participation in the research.

1. Schwartz JE, Stone AA. The analysis of real-time momentary data: a practical guide. In: Stone AA, Shiffman S, Atienza AA, Nebeling L, eds. *The science of real-time data capture: self-reports in health research*. 1st ed. New York: Oxford University Press; 2007: 76-116.
2. Dilorio C, Dudley WN, Soet JE, McCarty F. Sexual possibility situations and sexual behaviors among young adolescents: the moderating role of protective factors. *J Adolesc Health*. 2004; 35(6): 528-e511.
3. O'Donnell L, Myint-U A, O'Donnell CR, Stueve A. Long-term influence of sexual norms and attitudes on timing of sexual initiation among urban minority youth. *J Sch Health*. 2003; 73(2): 68-75.
4. Santelli JS, Kaiser J, Hirsch L, Radosh A, Simkin L, Middlestadt S. Initiation of sexual intercourse among middle school adolescents: the influence of psychosocial factors. *J Adolesc Health*. 2004; 34(3): 200-208.
5. Shiffman S, Stone AA, Hufford MR. Ecological momentary assessment. *Annu Rev Clin Psychol*. 2008; 4: 1-32.
6. Mehl MR. Eavesdropping on health: a naturalistic observation approach for social health research. *Soc Personal Psychol Compass*. 2007; 1(1): 359-380.
7. Dennison L, Morrison L, Conway G, Yardley L. Opportunities and challenges for smartphone applications in supporting health behavior change: qualitative study. *J Med Internet Res*. 2013; 15(4), e86.
8. McGillicuddy JW, Weiland AK, Frenzel RM, et al. Patient attitudes toward mobile phone-based health monitoring: questionnaire study among kidney transplant recipients. *J Med Internet Res*. 2013; 15(1), e6.
9. Krueger RA, Casey MA. *Focus groups: a practical guide for applied research*. 4th ed. Thousand Oaks: SAGE Publications; 2014.
10. Stewart DW, Shamdasani PN. *Focus groups: theory and practice*, vol. 20. Thousand Oaks: Sage Publications; 2015.
11. Maguire M. Methods to support human-centred design. *Int J Hum Comput Stud*. 2001; 55(4): 587-634.
12. Nielsen. Smartphone switch: three-fourths of recent acquirers chose smartphones. 2013; http://www.nielsen.com/us/en/insights/news/2013/smartphone-switch–three-fourths-of-recent-acquirers-chose-smart.html. Accessed December 24, 2014.
13. Nielsen. Multiplying mobile: how multicultural consumers are leading smartphone adoption. 2014; http://www.nielsen.com/us/en/insights/news/2014/multiplying-mobile-how-multicultural-consumers-are-leading-smartphone-adoption.html. Accessed Dec 24, 2014
14. Pew Research Internet Project. Mobile technology fact sheet: highlights of the Pew Internet Project's research related to mobile technology. 2014; http://www.pewinternet.org/fact-sheets/mobile-technology-fact-sheet/. Accessed Dec 24 2014
15. Conner TS. Experience sampling and ecological momentary assessment with mobile phones. 2015; http://www.otago.ac.nz/psychology/otago047475.pdf. Accessed June 29 2015
16. Ritter S. Apple's ResearchKit development framework for iPhone apps enables innovative approaches to medical research data collection. *J Clin Trials*. 2015; 5, e120.
17. *The connected enterprise: keeping pace with mobile development*. Framingham, MA: CIO Strategic Marketing Services and Triangle Publishing Services Co. Inc.;2014.
18. Nielsen. Mobile millennials: over 85% of generation Y owns smartphones. 2014; http://www.nielsen.com/us/en/insights/news/2014/mobile-millennials-over-85-percent-of-generation-y-owns-smartphones.html. Accessed Dec 24, 2014.
19. Shiffman S, Hickcox M, Paty JA, Gnys M, Kassel JD, Richards TJ. Progression from a smoking lapse to relapse: prediction from abstinence violation effects, nicotine dependence, and lapse characteristics. *J Consult Clin Psychol*. 1996; 64(5): 993-1002.
20. Shiffman S, Scharf D, Shadel W, et al. Analyzing milestones in smoking cessation: an illustration from a randomized trial of high dose nicotine patch. *J Consult Clin Psychol*. 2006; 74: 276-285.
21. Ritter FE, Baxter GD, Churchill EF. User-centered systems design: a brief history. *Foundations for designing user-centered systems*: Springer; 2014:33–54.

22. Larman C, Basili VR. Iterative and incremental development: a brief history. *Computer*. 2003; 36(6): 47-56.
23. Beck K, Beedle M, van Bennekum A, et al. Manifesto for Agile Software development. 2001; http://agilemanifesto.org/. Accessed Dec 24 2014.
24. Cohen D, Lindvall M, Costa P. An introduction to agile methods. *Adv Comput*. 2004; 62: 1-66.
25. Royce WW. Managing the development of large software systems. Paper presented at: proceedings of IEEE WESCON1970.
26. Sommerville I. *Software engineering*. 10th ed. Boston: Addison-Wesley; 2015.
27. Boehm B. Get ready for agile methods, with care. *Computer*. 2002; 35(1): 64-69.
28. Abrahamsson P, Warsta J, Siponen MT, Ronkainen J. New directions on agile methods: a comparative analysis. Paper presented at: Software Engineering, 2003. Proceedings. 25th International Conference on 2003.
29. Dybå T, Dingsøyr T. Empirical studies of agile software development: a systematic review. *Inf Softw Technol*. 2008; 50 (9): 833-859.
30. Munassar NMA, Govardhan A. A comparison between five models of software engineering. *IJCSI*. 2010; 5: 95-101.