Adventure Code Camp: Library Mobile Design in the Backcountry

David Ward ,
James Hahn,
and Lori Mestre

ABSTRACT

This article presents a case study exploring the use of a student coding camp as a bottom-up mobile design process to generate library mobile apps. A code camp sources student programmer talent and ideas for designing software services and features. This case study reviews process, outcomes, and next steps in mobile web app coding camps. It concludes by offering implications for services design beyond the local camp presented in this study. By understanding how patrons expect to integrate library services and resources into their use of mobile devices, librarians can better design the user experience for this environment.

INTRODUCTION

Mobile applications offer an exciting opportunity for libraries to expand the reach of their services, to build new connections, and to offer unique, previously unavailable services for their users. Mobile apps not only provide the ability to present library services through mobile views (e.g., the library catalog and library website), but they can tap into an ever-increasing list of mobile-specific features. By understanding how patrons expect to integrate library services and resources into their use of mobile devices, librarians can better design the user experience for this environment. By adjusting the normal app production workflow to directly involve students during the formative stages of mobile app conception and design, libraries have the potential to generate products that more accurately anticipate real-life student needs. This article details one such approach, which sources student talent to code apps in a fast-paced, collaborative setting.

As part of a two-year Institute of Museum and Library Services (IMLS) grant, an academic library–based research team investigated three different methods for involving users in the app development process—a student competition, integrated computer science class projects, and the coding camp described in this article. The coding camp method focuses on a trend in mobile software development of having intensive two-to-three-day coding events that result in working prototypes of applications (e.g., iPhoneDevCamp, http://www.iphonedevcamp.org/). Coders typically work in groups to simultaneously learn how new software works, and also develop a functioning app that matches an area of personal interest. Camps promote collaboration, which provides additional networking and social outcomes to attendees. Additionally, camps provide an

David Ward (dh-ward@illinois.edu) is Reference Services Librarian, **James Hahn** (jmhahn@illinois.edu) is Orientation Services & Environments Librarian, and **Lori Mestre** (lmestre@illinois.edu) is Head, Undergraduate Library and Professor of Library Administration, University of Illinois at Urbana-Champaign.

opportunity for software makers to promote their services and products, and they can result in new code and ideas on which to base future products.

For academic libraries, a camp environment provides an educational opportunity for students, particularly those in a field with a computing or engineering focus, to learn new coding languages and techniques and to gain experience with a professional software production process that runs the full timeline from conception to finished product. Coding camps offer a chance for librarians to get direct student feedback on their own software development goals. The resulting applications provide potential benefits to both groups—students have a functional prototype to enhance their classroom experiences and a codebase to build on for future projects, and the librarians gain an insight into students' desires for the content of mobile apps, code to integrate into existing apps, and direct student input into the iterative design process.

This article presents the results of a mobile application coding camp held in fall 2013. The camp method was tested as a way to explore a less time- and staff-intensive process for involving students in the creation of library mobile apps. Three specific research questions framed this investigation:

1. What library and course-related needs do students believe would benefit from the development of a mobile application?
2. Is the library providing access to data that is relevant to these needs, and is it available in a format (e.g., Restful APIs) that end users can easily adopt into their own application design process?
3. How viable is the coding camp method for generating usable mobile app prototypes?

LITERATURE REVIEW

In line with efforts in academic libraries to operationalize participatory design for novel service implementation,[1] the library approach to code camps included sourcing student technical expertise in line with other tech companies' approaches to quickly iterating prototypes that may advance or enhance company services. While coding camps happen in corporate settings, other types of camps try to publicize technologies, like a programming language, while others still are directed toward a specific cohort.[2] The departure point for the library was in understanding other ways the library might consider organizing and pairing its resources of APIs with other available campus services.

A few highly visible and notable corporate "hackfests" or "hackdays" include the Facebook hackdays, in which Facebook timeline was developed (http://www.businessinsider.com/facebook-timeline-began-as-a-hackathon-project-2012-1). The mobile app company Twitter also has monthly hackfests where employees from across the company work for a sustained period (a weekend or Friday) on new ideas putting together prototypes that may transition into new services for the company.

An example of code camps from academia are the MHacks camps at the University of Michigan (http://mhacks.challengepost.com/), among the largest code camps for university students in the Midwest. These camps are notable for their funding from corporations and for their support of student travel from colleges around the country to participate at the University of Michigan. At each event, coders are encouraged to make use of the corporate APIs that student programmers may make use of once they graduate or form companies after graduation.

On the professional front, digital library code meet-ups (such as that of the code4lib pre-conference: http://code4lib.org/) are an opportunity for library technologists to share strategies and new directions in software using hands-on group coding sessions that last a half or full day. A recent digital event for the Digital Public Library of America (DPLA) hosted hackfests to demonstrate interface and functional possibilities with the source content in the DPLA. Similarly, the Hathi Trust Research Center organized a developer track for API coaching at their conference so that participants would have hands-on opportunities to use the Hathi Trust API (http://www.hathitrust.org/htrc_uncamp2013).

Goals of coding camps include development of new services or creation of value-added services on already existing operations. Code is not required to be complete, but functional prototypes help showcase new ways of approaching problems or novel solutions. Recently, MHacks issued the call to form new businesses at their winter hackathon (http://www.mhacks.org). Libraries are typically less interested in new businesses, but rather seek new service ideas and new principles for organizing content via mobile and to do so in such a way that will source student preferences for location specific services, a key focus for the research team's Student/Library Collaborative IMLS grant.

METHOD

While the camp itself took only two days, there was a significant amount of lead-time needed to prepare. In addition to obtaining standard campus institutional-review-board permissions for the study, it was also necessary to consult the Office of Technology Management to devise an assignment agreement covering the software generated by the camp. The research team chose a model that gave participating students the option to assign co-ownership of the code they developed to the library. This meant that both students and the library could independently develop applications using the code generated during the camp.

Marketing for the camp specifically targeted departments and courses where students with interest and skills for mobile application development were likely to be found, particularly in computer science and engineering. Individual instructors were contacted, as well as registered student organizations, to help promote the camp. Attendees were directed to an online application form, where they were asked to provide information on their coding skills and details on their interest in mobile application development.

Ten students were ultimately selected from the pool and, of those, six attended the camp. A pre-camp package was sent to these students to help them prepare for the short, intense timeframe the event entailed. This package included details on library data that were available to base applications on through web APIs, as well as brief tutorials on the coding languages and data formats participants needed to be familiar with (e.g., JavaScript, JSON, XML, etc.). Participants were also provided with information on parking and other logistics for the event.

The research team consisted of librarians and academic professionals involved in public services and mobile design, and student coders employed by the library to serve as peer mentors. The team designed the camp as a two-day experience occurring over a weekend (Friday evening to Saturday late afternoon). The first day was scheduled as an introduction to the camp, with details on library and related APIs that could be used for apps and an opportunity for participants to brainstorm app ideas and form design teams. The day ended with some preliminary coding and consultation with camp organizers about planned directions and needs for the second day.

The second day of the camp mostly for coding, with breaks scheduled for food, presentations of work-in-progress, and an opportunity to ask questions of the research team. The day ended with each team presenting their app, describing their motivation in designing it and the functionality they had been able to code into it.

Given the brief turnaround time, the research team put a heavy focus during the orientation session on clearly articulating the need to develop apps germane to student library needs. Examples from the student mobile app design competition conducted in February 2013 were provided as starting points for discussion, as these reflected known student desires for mobile library applications.[3]

After the camp ended, students who elected to share their code with the library were given details on how and where to deposit the code. Post-camp debriefing interviews (lasting 30 to forty-five minutes each) were scheduled individually with all participants to get their feedback on the setup of the event as well as what they felt they learned from the experience.

DISCUSSION

Researcher observations and feedback from students, both during the camp and in individual interviews afterwards, led to several insights about what sorts of outcomes libraries might anticipate from running camps, how to best structure library coding camps, what outcomes students anticipate from participating in a sponsored camp environment, and what features and preferences students have for mobile apps designed to support their academic endeavors.

A key student assumption, which emerged from comments at the event and through subsequent student interviews, was that students anticipated completing a fully functioning mobile app by the end of camp. Instead, the two student teams each finished with an app that, while it included some of the features they desired, still required additional coding to be fully realized. Several suggestions were made for how this need might be met at future events. The most consistent feedback from the students was that they would have liked an additional day of coding (three total camp days), so that they could have gotten further on the implementation of their app ideas. During the exit interviews, one student noted that the two-day timeframe really only allowed for sketching out an idea for an app, not coding it from scratch.

A pair of related suggestions from students included having templates for mobile apps available to review to get up to speed on new frameworks (particularly jQuery), and secondly, a longer meet-and-greet for teams prior to beginning work during which they could compare available coding skills and have some extended brainstorming of app ideas. Students were somewhat mixed in their desire for assistance in developing app ideas—some appreciated the open-endedness of the camp, but others wanted a more organizer-driven approach.

Some students suggested having time to work with library staff after the camp to finish or polish their apps. This observation suggests the enthusiasm students had for the camp itself, and specifically for having a social, structured, and mentored opportunity to develop their coding skills. Based on these requests, the research team created “office hours” on the fly after the camp ended to support this request. Research team members and coding staff communicated times when team members could come into the library and get additional help with developing their apps.

The students had very similar themes for app features to those that the research team observed in an earlier student mobile app competition study. Notable categories included the following:

- Identify and build connections with peers from courses.
- Discover campus resources and spaces.
- Facilitate common activities such as studying and meeting for group work.

Students remarked that the camp was an opportunity to both meet people with similar coding interests as well as to learn more about specific functional areas of app development (specific coding languages, user interface design, etc.) in which they had little experience. JQuery and JavaScript for user-facing designs were particular areas of interest. Many students had some in-depth background working on pieces of a finished software product but had not previously done start-to-finish software design; this was a big selling point for the camp. The collaborative nature of the camp also matched students’ preferences to work in teams and to learn from peers. While the research team had coders on hand to assist with both the library APIs, as well as jQuery basics, most teams did the majority of their work themselves, and preferred self-discovery. Each team did eventually ask for coding advice, but this occurred toward the end of the camp, once their apps were largely coded and they needed assistance overcoming particular sticking points.

The other piece of advice students asked organizers about concerned identifying APIs for locations of campus maps, and other related resources to serve as data sources powering their apps. In the course of assisting with these requests, researchers discovered another key issue facing library mobile app development—the lack of campus standards for presenting information across different colleges and departments. In particular, maps of rooms inside campus buildings were not provided in a consistent or comprehensive way. This was particularly frustrating to the team that was attempting to develop an app featuring turn-by-turn navigation and directions to meeting rooms and computer labs.

In addition to sharing information on known APIs and data sources, camp organizers also learned about previously unknown data sources from the student teams. One example was a JSON feed for the current availability of computers in labs provided by the College of Engineering. While this feed was beneficial to starting work on an app for one team, it also led to frustration because feeds for other campus computer labs did not exist, and the team was limited to designing around the specific labs that did have this information available.

Observed student discussions about the randomness of data availability also highlighted one of the key themes of student-centered design—the conceptualization of a university as a single entity, the various parts of which combine and come in and out of focus depending on the current student task. Related student feedback from one of the post-event interviews described a strong desire to create integrated, multifunction apps to meet student needs as opposed to a variety of apps that each did one thing. The siloed nature of campus infrastructures frustrates this desire to some extent but also creates opportunities for students to build a tool that meets a real need among their peers to comprehend and organize their academic environment. This observation also matches those found during the aforementioned student competition.

CONCLUSION AND FUTURE DIRECTIONS

Student feedback on the camp, as a whole, was very positive, and in the individual interviews, students noted they would like to participate in another camp if it was offered. On the library side, the research team felt that the camp was useful to their ongoing mobile app development process, partially for the code generated but primarily for the direct feedback on what types of apps students wanted to see. The start-up time and costs for the project were low, as expected, and the insights into student mobile preferences seemed proportionate to this outlay. The camp method should be reproducible in a variety of library environments. The key assets other libraries will need to have in place to run a camp include staff with knowledge of client-side API use (in particular jQuery, CORS, or related skills), and knowledge of campus data sources that students may wish to pull from. Third-party APIs with bibliographic data (e.g., Good Reads) could also be used as placeholders for libraries that do not have access to APIs for their own catalogs or discovery systems.

Student suggestions for extending the camp by a day, and their ideas for how to structure it for student success, were very specific and actionable and provided excellent guidance. One of their ideas was to develop tutorials and templates that could be introduced as a pre-camp meeting. This would not add too much prep time. Another idea for a future camp would be to develop a specific theme for teams, which would allow for more documentation of and practice with specific APIs.

The low attendance was a concern, so for the next camp twice the number of desired participants will be invited to ensure both a variety of coding skills and interests as well as opportunities for more teams to be formed. Additionally, partnerships with student coding groups or related classes should help to drive up attendance.

The biggest difficulty moving forward will be developing campus standards for data that can be made available to students about resources, spaces, and services. As noted above, students typically do not design a “library app,” rather they look to build a “student app” that pulls in a variety of data from across campus. Functions of apps are therefore more oriented toward common student activities like studying, socializing, and learning. A related challenge will be to provide adequate format and delivery mechanisms for access to supporting data feeds.

Cognizant of the silo issue, noted above, as libraries present their own data for student consumption, these tendencies towards a unified view need to be taken into account. Completion of an assignment is more than identifying three scholarly sources; it might involve identifying a space to do the research, locating peers or mentors for either the research or writing process, locating suitable technology to complete an assignment, and a variety of other needs. The features and information presented on a library’s website should be designed as modular building blocks that can fit into other campus services in a similar way to how course reserves are sometimes presented in campus learning management services alongside syllabi and assignments. Separating library content (e.g., full-text articles, room information, research services) from library process can help with freeing information about what libraries have to offer and can facilitate broader discovery of services and resources at point of need. Key to this process is recognizing the student desire to shape the resources they need into a comprehensible format that matches their workflow rather than forcing students to learn a specific, isolated, and inflexible path for each part of the projects they work on.

This study has shown that a collaborative process in technology design can yield insights into students’ conceptual models about how spaces, resources, and services can be implemented. While the traditional model of service development often leaves these considerations until the very end in a summative assessment of service, the coding camp and collaborative methods presented here provide librarians a new tool for adding depth to service design and implementation, ultimately resulting in services and platforms that are informed by a more well-rounded and deeper understanding of the student mobile-use experience. In that regard, the initial research questions that framed this study could also be used by other libraries as they explore the library and course-related needs that students could benefit from with the development of mobile applications, as well as to determine if their library provides access to data that is relevant to those needs. The results from this study have affirmed that, for at least the library from this study, that the coding camp method is viable for generating usable mobile app prototypes. It also affirmed that by directly involving students during the formative stages of mobile app conception and design, the products of those apps more accurately reflect real-life student needs.

REFERENCES

1. Council on Library and Information Resources (CLIR), *Participatory Design in Academic Libraries: Methods, Findings, and Implementations* (Washington, DC: CLIR, 2012), http://www.clir.org/pubs/reports/pub155/pub155.pdf.

2. "Hackathon," Wikipedia, 2014), http://en.wikipedia.org/wiki/Hackathon.

3. David Ward, James Hahn, and Lori Mestre, "Designing Mobile Technology to Enhance Library Space Use: Findings from an Undergraduate Student Competition," *Journal of Learning Spaces* (forthcoming).