Prototyping iPhone Apps: Realistic Experiences on the Device

Anders P. Jørgensen
AU-HIH
anders@anderspj.dk

Matthijs Collard
UNITiD
matthijs@unitid.nl

ABSTRACT

In this paper we evaluate Touch Application Prototype - a tool for designers to quickly create interactive and realistic prototypes of Apple® iPhone® apps and test them on the device. We define 5 requirements such as Speed, Practicality and Realism, and evaluate the tool during the development of a mobile work tool. Users intuitively use their inherent knowledge about touch interfaces, revealing expectations towards the use of gestures, and testing the interface's affordance. TAP rivals the speed and ease of paper prototyping, yet offers a realistic look and feel, without any coding. It is offered as a public, free tool.

Keywords

Mobile applications, graphical user interfaces, prototyping, user evaluation, iPhone, iPad.

INTRODUCTION

In our work with designing iPhone apps, our designers have found the need for a prototyping tool that is easier to use than Apple's Xcode® and Interface Builder, yet offers a higher degree of realism than regular click-through prototypes, and enables testing of multiple concepts in short time and at low cost. This has led to the development of TAP - Touch Application Prototype.

This paper starts by defining 5 requirements for an early-stage prototyping tool. We then argue why they are relevant based on literature. TAP is then described, followed by a case story of the tool being used in a commercial project. Finally, the learnings are discussed and suggestions are made for improvements.

Requirements

A prototyping tool for use by designers in the conceptual iPhone app design phase should be: (1) quick and easy to make, (2) practical to test in users' real life settings, (3) displayable on the actual hardware, (4) accessible without requiring the presence of a facilitator and (5) enable realistic fidelity and transitions.

In the early stage of a project, the weight should be put on ideation and ensuring the concept is useful, or "getting the right design" according to Buxton (2007)[3]. This requires multiple iterations to be rapidly tried out, to zero in on the best path. In addition, user evaluation of mobile systems should be done in real-life. Even though this requires more work than staying "in the lab", it has proven to provide the most relevant feedback about users' work flow and use situations [5, 12]. Buxton (2007)[3] argues that initial prototypes should be low fidelity, but using e.g., paper prototypes in the wild has proven impractical [13]. Paper prototypes and other simplistic methods also usually require a facilitator to manually perform the interface's state changes. But having designers and researchers tagging along can be obtrusive and influence results negatively [13]. Research has found that less obtrusive methods produce just as valuable, and in many cases, more honest results [6, 9]. Therefore, users should be able to try out prototypes on their own, or at least without constant interference.

Bolchini, Pulido and Faiola (2009)[1] describe a method where hand drawn sketches of the interface are digitized and transferred to the iPhone's photo album, allowing users to flick (finger gesture [14]) through screen states on the device. Their method lives up to our requirements 1, 2, 3 and 4, but is weak on number 5. Although the interface is displayed on the real device, it can only be navigated linearly using the iPhone's built in image viewer. There is also no way of clicking buttons or scrolling lists, so common ways of navigating apps are replaced by the wrong input (flick horizontally to advance photo) and visual feedback is limited to sideways sliding of the screen content. Transitions and animations are crucial to the user experience [3], and using the correct physical metaphors on touch screen devices significantly influences users' effectiveness [8]. So in this particular area Bolchini et al.'s method is actually worse than manually operated physical paper prototypes.

In addition, and contrary to Buxton's promotion of sketch-like fidelity, research comparing high- and low-fidelity prototypes has found that more realistic prototypes can actually result in more valuable feedback [10]. For users who are not used to working with sketches, the lack of a polished interface can be confusing and result in overly negative feedback [4, 13]. If users have become accustomed to the high standards of the iPhone interface, the evaluation of a prototype that looks "wrong" might be misleading. Devices like the iPhone also have limited screen real estate, making it important for designers to be sure that their proposed designs will actually fit on screen, and that touch areas are large enough to be operated with fingertips [14]. A prototyping tool should therefore enable realistic interfaces.

TAP - TOUCH APPLICATION PROTOTYPE

TAP falls into the category of “Smoke-and-Mirrors”, where technology is used to create the illusion of a working product [3]. With TAP we use the iPhone’s web browser to display what is essentially a mini website with clickable images of the application interface. Having a click-through prototype is not so special in itself. But what makes this tool interesting for designers, is that without any coding, they can make a prototype that:

- Runs full screen without the default Safari browser navigation at the top and bottom of the screen.
- Animates transitions between screens with effects like cube, dissolve, flip, pop, slide-up and swap.
- Cashes the prototype on the iPhone, so it loads instantly and responds as snappy as a native application.
- Integrates video playback and animated images.
- Allows the designer to lay out the whole interface in Fireworks, a program likely familiar to many designers, and to set the rest up through a simple web interface.

Technically, TAP is a library of files containing custom developed PHP and jQuery code that makes the prototype come to life.

Building the Prototype

First of all, designers may start out with which ever method they prefer for ideation and conceptualization, be it paper prototypes, stencil based wireframes, foam props, etc. When one or more app concepts are ready for testing, the interface has to be digitized using Adobe’s® Fireworks®. In our experience, the easiest way is to rebuild the interface in Fireworks using a stencil containing the iPhone default user interface elements. Many such stencils are available for free online [7]. We recommend using a realistic but neutral design following Apple’s Human Interface Guidelines (Available from Apple to registered developers).

Every application state should be made as an individual page with so called hotspots (figure 1). A hotspot is an active area that links to a new page when the user taps it. Each hotspot can be assigned an animated transition. Meanwhile, the TAP folder containing code files and instructions can be downloaded at: http://unitid.nl/tap.

When all screens are made in Fireworks and linked together, the project is exported in Dreamweaver Library (.lbi) format, into the /Library/ subfolder of the TAP package, and everything is uploaded to a web server.

When visiting *http://yoursite.com/*tap/build with a desktop browser, a web interface offers simple setup in a few steps. When everything is set up, just hit the “Build prototype button, and the prototype is ready. At first launch, the user is prompted to save a shortcut on their Home Screen, making it appear as a real app on the phone.

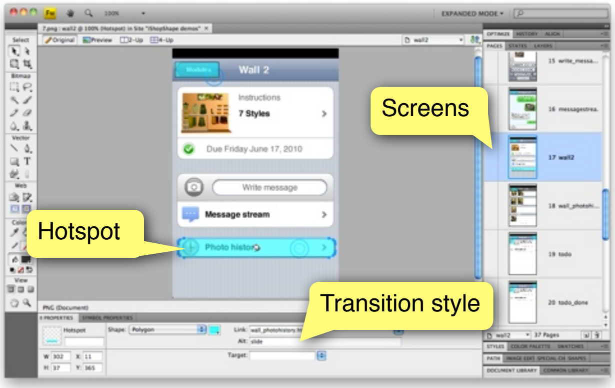

Figure 1: A TAP prototype in Adobe Fireworks .

Because the entire prototype resides on the web, it is very easy to share with many users, even remote users, as long as they have an iPhone and an Internet connection. There is no need to collect device IDs and distribute provisioning profiles as in regular beta testing. Hence, users need minimal instructions and can play around with the prototype on their own phone as if it were a real app.

We recommend that designers only create the states and transitions necessary to provide the desired experience and get valuable feedback. Trivial functions like text entry can be simplified so that one tap anywhere on the keyboard completes a text field, and secondary features can be left out, or remain passive. In some cases we have used visual queues to show users which areas can be tapped and which cannot, for instance with red outlines, elements grayed out, or even small info texts.

EVALUATING TAP

In Buchenau and Suri’s article on Experience Prototyping they describe how prototypes can be used not only for “Exploring and evaluating design ideas”, but also for “Communicating ideas to an audience” [2]. We have found TAP very valuable for both.

The “audience” - our clients - have in most cases been given a version with markings on the active buttons. This way, they could simply be sent a link and explore the prototype on their own, but without being frustrated from pressing “dead” buttons. This approach has also been used with users in the early stages when the design is very tentative and only a few buttons are wired up.

It has been easy to test multiple concepts up against each other by setting up an intro screen with explanations and links to different prototypes. This way, users still only had to visit one URL with their phone.

Evaluating a work tool

Here we present some of the qualitative learnings gained during a concept study for an app to be used by employees in fashion retail stores. Based on interviews conducted in their real life setting, we presented an early sketch with the look of a real app. We clearly explained that it was an early draft, and that they should feel free to criticize and suggest additions.

Users were surprisingly quick to pick up on the navigation of the app, clicking, scrolling and exploring. Some users owned an iPhone themselves, others not. The seemingly immediate familiarity however was apparent for all users. In contrast to what might be expected with a realistic prototype, there were only few comments on design details. These were mostly in the category of "I like the icons...", but there were no comments on the default blue-toned iPhone UI elements. It seems that because the iPhone has gained so much public exposure, using the default interface elements is perceived as "neutral", whereas a deliberately sketchy style would stick out.

The realistic interface turned out to have an unexpected side effect. As a user was looking at an image in the prototype, she suggested that maybe it should be possible to zoom in. While saying this, she did the pinch-to-zoom gesture on the screen completely unaided (figure 2).

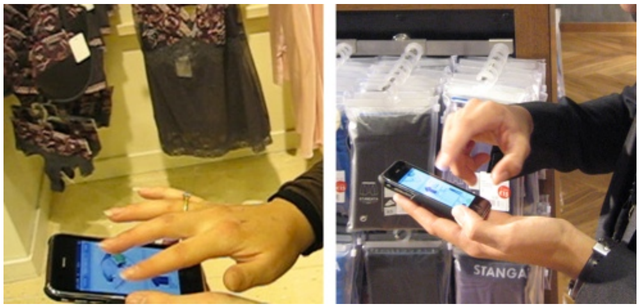

Figure 2: Left: User demonstrating pinch-to-zoom gesture as a way to enlarge photos. Right: User scrolling list of images and text.

In a later iteration we decided to explore this path in the design of a calendar module. With simple graphic cues, we made it look as if some content was hidden off the side of the screen (figure 3). When users came to this screen, they intuitively tried swiping sideways. This was not possible in the prototype, instead users were told they had to press small arrow keys. Even after knowing this, they kept trying to swipe when they forgot the tool's limitations. In some cases users were frustrated that they couldn't use more gestures. The transitions and effects that the prototype *was* able to do, were however explicitly praised by testers as being very realistic. The dialogue with the facilitator was important in clarifying which limitations were a consequence of the prototyping tool.

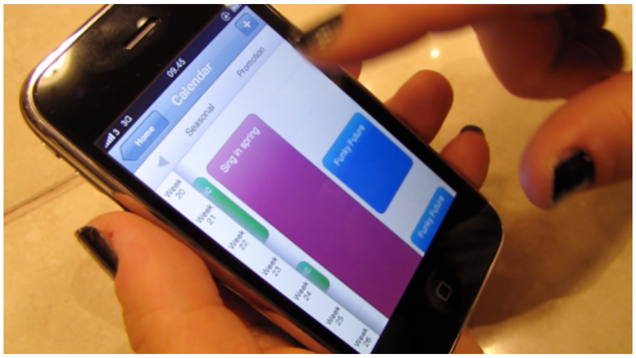

Figure 3: Users kept trying to swipe sideways, confirming the power of visual suggestions, and users's expectations.

One of the users (A) was particularly enthusiastic, and asked to have the prototypes on her iPhone, so she could explore it on her own during idle periods in the store. Another user (B) went on holiday during the test period, but wanted to still be involved. She was sent a link via SMS, and a few hours later she sent a message back with her thoughts. At a later session users A and B were interviewed together. To our surprise, A started explaining the app's detailed features to B, and had even memorized which buttons were active and which not, from exploring it on her own (this version was without markings on active buttons). A and B then went into a detailed discussion about features and integration of the work tool in their daily operations. User A also provided concrete ideas for shortcuts, based on interface elements familiar from the default phone module of iPhone.

This indicates that a realistic experience can invite users to start pretending the app is real, advancing feedback from the initial first-hand reactions, to more considered day-to-day use considerations. Realism may also aid associations to other apps, making users more comfortable in suggesting specific features.

DISCUSSION AND FURTHER DEVELOPMENT

The five requirements we set up for an early-stage prototyping tool for iPhone apps were: (1) quick and easy to make, (2) practical to test in users' real life settings, (3) displayable on the actual hardware, (4) accessible without requiring the presence of a facilitator and (5) enable realistic fidelity and transitions.

In our use of the tool, we have found that the speed of building the interface digitally and making it available online is comparable in speed to creating precise hand-drawn screens, scanning, cropping and linking them together. Being online, it is very easy to share with anybody with an iPhone, and users can view it wherever, whenever. Including instructions and highlighting active buttons reduces the user's need for assistance. So TAP meets requirements 1, 2, 3 and 4. Because the prototype is web based it has a number of limitations. Some gestures like dragging objects around the screen and pinch to zoom are not currently supported. It is not possible to simulate immersive content such as 3D games and augmented reality, although this type of content could be demonstrated through embedded video clips. Animated transitions, caching and loading screens are however very realistic. Thus, requirement 5 is met, provided the tool is used for non-immersive applications. For more complicated interactions, realism will begin to break down and the experience will become obviously "fake", requiring more explanations.

It was not expected that the prototype's realism would help uncovering user's tacit knowledge about gestures, and demonstrate which gestures they expected to be able to perform. Considering how affordance and visual cues can be used to explore gestures is an exciting area for further exploration.

It is important for us to stress that designers should never limit their creativity to fit the tool. For more advanced and novel interface concepts, Wizard-of-Oz techniques with a person simulating feedback [11], or simply programming parts of a real application, might be good supplements to TAP.

The tool's code and templates are free for anybody to modify and improve. One possible addition could be integration of web analytics tools, to add quantitative data about clicks and navigation patterns.

The iPhone platform is constantly evolving and offering new possibilities that help us improve the tool. The current version of TAP is by no means the ultimate tool, but it is the best middle ground solution we currently know of, bridging the gap between the speed and ease of passive sketches/images and the interactive, but resource heavy native programming.

REFERENCES

1. Bolchini, D., Pulido, D., and Faiola, A. "Paper in Screen" Prototyping: An Agile Technique to Anticipate the Mobile Experience. *interactions* (July / August, 2009), 29-33.
2. Buchenau, M., and Suri, J.F. Experience Prototyping. *DIS '00* (Brooklyn, NY, 2000), ACM, 424-433.
3. Buxton, B. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann, San Francisco, CA, USA, 2007.
4. Dehaes, C., and Nuyens, L. Cutting Edge Usability Design and Smashing Graphics: The Perfect Recipe for Firing up a Sophisticated Pharmaceutical Touch Screen Application. *CHI 2008* (Florence, Italy, April 5-10, 2008), ACM, 2095-2108.
5. Duh, H.B., Tan, G., and Chen, V.H. Usability Evalutation for Mobile Device: A comparison of Laboratory and Field Tests. *MobileHCI '06* (Helsinki, Finland, September 12-15, 2006), ACM, 181-186.
6. Isomursu, M., Kuutti, K., and Väinämö, S. Experience Clip: Method for User Participation and Evaluation of Mobile Concepts. *Proceedings Participatory Design Conference 2004* (Toronto, Canada, 2004), ACM, 83-92.
7. Kuwamoto, S. Fireworks toolkit for creating iPhone UI mockups (February 11, 2009). Available at http://blog.metaspark.com/2009/02/fireworks-toolkit-for-creating-iphone-ui-mockups/.
8. Lee, S.S., and Lee, W. Exploring Effectiveness of Physical Metaphor in Interaction Design. *CHI 2009* (Boston, MA, April 4-9, 2009), ACM, 4363-4368.
9. Leitner, M., Wolkerstorfer, P., Geven, A., Höller, N., and Tscheligi, M. Evaluating a Mobile Multimedia Application in Field Trials: the cost-benefit of self-report methods. *Mobile Living Labs 09* (Bonn, Germany, September 15, 2009), 27-30.
10. Lim, Y.K., Pangam, A., Periyasami, S., and Aneja, S. Comparative Analysis of High- and Low-fidelity Prototypes for More Valid Usability Evaluations of Mobile Devices. *NordiCHI 2006* (Oslo, Norway, October 14-18), ACM, 291-300.
11. Molin, L. Wizard-of-Oz Prototyping for Cooperative Interaction Design of Graphical User Interfaces. *NordiCHI '04* (Tampere, Finland, October 23-27), ACM, 425-428.
12. Nielsen, C.M., Overgaard, M., and Pedersen, M.B. It's worth the hassle! The Added Value of Evaluating the Usabiiity of Mobile Systems in the Field. *NordiCHI 2006* (Oslo, Norway, October 14-18), ACM, 272-280.
13. Sá, M., and Carriço, L. Lessons from Early Stages Design of Mobile Applications. *MobileHCI 2008* (Amsterdam, Netherlands, September 2–5, 2008), ACM, 127-136.
14. Saffer, D. *Designing Gestural Interfaces: Touchscreens and Interactive Devices.* O'Reilly Media, Sebastopol, CA, USA, 2008.