ENTERPRISE COMPUTING: APP DISTRIBUTION

STEPHEN GILMORE

1. App distribution

So, good morning everyone! it seems to have gone 9:15, so it's time to make a start on the content of the lecture. It's a great pleasure to see you all here.

The subject of the lecture today is how you distribute mobile phone apps. So, if you're a software developer making a mobile phone app at some point you need to put it out to testers. But how do you do that?

—

Most of us know about "app stores" where *once you have* a product that you are either going to sell, or give away, then you can get it onto an app store. But before you can do that you have to do *testing.* So how is that done?

—

The kind of app that I am going to look at is an *accessible* app trying to help blind or partially-sighted people. Designing an accessible app is *the most extreme form* of usability engineering.

You don't just want a "user friendly" app, you want something that is really *user helpful*, considering that many of the people operating the app *won't be able to see the screen of the phone.*

If that is your target audience, if those are the people you are designing for, if those are the people that you have in mind, then you've got a real challenge.

—

Most of us design apps to be operated by the eyes and the finger, but when designing for a blind person you are looking for an app to be operated by the ear and the finger. They are going to use VoiceOver[1]. They are going to *hear* the commands, rather than see the screen. And, of course, VoiceOver (as we saw) completely changes the conventional user-interface experience that we are all used to.

Things like *muscle memory* play a part here. As sighted developers[2] we are used to using phones (including smartphones) in the way that a sighted person does.

You look at the screen.

You put your finger down where you want it to go.

There is no exploring, or seeking.

You look. You touch.

—

So, if you are an implementor and you are testing your work both in the conventional way (because, of course, your app will be used by sighted people probably

School of Informatics, The University of Edinburgh.

[1] Or TalkBack on Android, and so forth, but this case study is considering a particular iPhone app being developed by Transport for Edinburgh.

[2] All the people at this lecture were sighted.

as well, or at least partially-sighted people) but you also want it to work for *blind people* then you have a lot of work cut out for you.

There will be the usual kinds of testing. There will be *interactive testing* (where you use the app in User Mode, as a user would use it). There will be *unit testing* (automate-able tests that you run after every change to the code of the app). And so forth, whatever kinds of testing that you like.

But, then, you need to switch into a completely different mode. You need to try using the app with VoiceOver.

Then with VoiceOver you don't really have *as much* automated testing support. You are now back to exploring the app with your finger, and your ear, and your eye. But the conventional user-interface experience has been completely changed.

—

Now, most software developers are sighted and VoiceOver buttons that a blind person "looks" for are literally invisible to the eye of a sighted person. (You find them with your finger, exploring the screen.)

So if you have a bug in your program[3] and you have a button in the wrong place (say), as far as VoiceOver is concerned, you **literally can't see it by looking at the screen**.

You have to use the app in VoiceOver mode and try to find it (the mis-placed VoiceOver button).

—

So, basically, the message here is that software developers trying to create *accessible apps*[4] need all of the assistance that they can get.

—

So, if you have a smartphone, or you've seen someone use a smartphone, we are all familiar with the idea of installing apps from the "app store". You get the apps from there and put them on your phone, so that they are coming from a trusted source. That's a pretty good distribution mechanism for software.

Someone has a look at the app beforehand, sees that it does not have any "spyware", viruses, and so on. If they think that the app is OK[5] then they put it up on the "app store" and say "OK, this app is now available for download."

—

The *Apple App Store* works in that way. Their apps are scrutinised carefully by Apple's own developers[6]

The *Google Play Store* is a bit more relaxed. It is more *open*, which is good but you don't have the same levels of guarantees of scrutiny.

There is the *Windows Phone Store*, and so on. There are others as well.

—

When you are at the stage of *user testing* (before app store approval, or "application store" approval, or whatever you call it) you have to distribute your app.

At that stage, apps are distributed using *archive files.*

We are all familiar with ZIP files, TAR file, and things like that (compressed archive files). An iOS archive file is literally a ZIP file under another name.

[3] Bugs are not *rare.* Incidentally, Grace Hopper popularised the term "debugging".

[4] And, by the way, shouldn't they *all* be accessible?

[5] They might make a mistake, but at least they try to check the app

[6] By the way, this is not solely automated scrutiny. Their developers will try your app. A real live human tries your app.

So, if you are downloading the new version of the Transport for Edinburgh[7] app designed for blind or partially-sighted people and you download it onto your computer then what you get is a "`.ipa`" file.[8]

So, what is an IPA file?

The "I", of course, stands for iOS. The "PA" I have no idea what it stands for[9]. The "A" is probably for "archive". An IPA file is an "iOS application archive file". It stores an iOS app, usually encrypted, and compressed.

It includes a binary for the architecture of the phone. Which, for Apple iPhones, is ARM.

It can only be installed on an iOS device.

—

OK, but files with the `.ipa` extension *are just ZIP files under another name.*

The other name is *helpful* because they can be recognised and treated differently from a routine ZIP file. The different name is "metadata" if you like.

"This is a *certain kind of ZIP file*," they are saying.

But, IPA files can be uncompressed by changing the extension to `.zip` and then unzipping in the usual way.

—

So, let's do that with the Transport for Edinburgh (TFE) Talk app and see what happens. That sounds like fun!

Well, you get a directory structure, as you would imagine.

We won't look at it in detail, but let's just skim it a bit.

- There is a top-level directory called "Payload" which is the content of the ZIP file.
- Inside that is a folder called "TalkingBuses.app".
- Inside that are two sub-folders.
  - One is called `.monotouch-32` (for 32-bit architectures).
  - The other is called `.monotouch-64` (for 64-bit architectures).
- Inside those are the sorts of things that you would expect. DLLs, and so on. `System.Core.dll`, `System.Xml.dll`, and, an EXE file, an executable file: `TalkingBuses.exe`.

Normally we don't see these sorts of things. A development environment such as Xcode packages this all up for you (you just get the ZIP file) but you can unzip it if you want to. And when you do, you find (files such as) DLLs. These are all binaries, it is not as though we are able to recover someone's source code by doing this. There are a few other things as well. The user interface of the app is designed with a drag-and-drop interface builder which produces `.nib` files. There is a database. There are image files in PNG[10] format. What else? There are

[7] Previously, "Lothian Buses".

[8] Not to be confused with India Pale Ale, an alcoholic drink served in public houses throughout Edinburgh, and lesser parts of Scotland.

[9] The built-in Apple dictionary under Mac OS X "Mavericks" says the following for IPA *abbreviation.*

- *India pale ale, a type of light-coloured beer similar to bitter. [said to have been brewed originally for the British colonies.]*
- *International Phonetic Alphabet.*

Perhaps Apple don't know what IPA stands for either.

[10] Portable Network Graphics

parameter list files (`.plist` files). There are some fonts, and a text file called `NOTICE`.[11]

—

So, if you want to have a little look inside this IPA file before you install it on your phone then you can. Because, after all, it's only a ZIP file.

2. Testing a beta version of a mobile phone app

Transport for Edinburgh are trying to recruit testers to test the new beta version of their app. The new app is a bit different from the standard one because it is designed especially for blind or partially-sighted passengers. The app is for iOS 8, or later. It has been developed by Lothian Buses PLC.

—

I will now take you through the process of how you get to the point where you could install this app on your device if you wanted to test it for Transport for Edinburgh. This is like an example, or a case study for this course, but please bear in mind that what we are trying to understand is **what is the process that needs to be gone through[12]?** What are the major steps, and why are they important?

—

The accompanying slides illustrate this with screenshots (which are literally a walk-through) but please think about *why we think that these steps are being done in this order.*

1. The first page is like an advertising page. It says "Recruitment" at the top. It wants to recruit *testers* to test the next version of the app.
2. You might, or you might not, be allowed to test the next version of the app. You have to *apply* to test it.
3. You can click a button and the developer of the app will receive your request.
4. If the developer chooses to accept your request then you are allowed to go further.
5. Once the developer has accepted your request, you will automatically see the app on your dashboard.
6. Suppose you press on "Apply". Then, what happens? When you press the "Apply" button an apply message is sent saying "I am volunteering to be a tester, or at least I am interested in doing so."
7. Then you get a message reading *Please wait: the developer needs to add your device.*[13] *You cannot install the app until you receive an email letting you know that a build is available. The developer still needs to add your device to the app's profile.*

[11] It is a copyright notice reading "Xamarin built applications contain open source software. For detailed attribution and licensing notices, please visit `http://xamarin.com/mobile-licensing`". The Xamarin Platform allows developers to *write their apps entirely in C#, sharing the same code on iOS, Android, Windows, Mac, and more* (paraphrasing the wording which is currently found on the `http://xamarin.com` home page). It was a bit surprising that there were DLLs and an EXE file in the IPA archive, but that clears that mystery up nicely. It seems that Transport for Edinburgh intend to make their new app for blind and partially-sighted passengers available on as many platforms as possible. *Well done, Transport for Edinburgh!!* (Transport for London, please take note.)

[12] . . . to install a beta version of a app on your mobile device.

[13] This is your hardware device, such as an iPhone, iPod or iPad, as appropriate for iOS devices.

That is the first thing that happens. You say "I am interested" and then you may (or may not) get an email back. And if you do, you can go further.

—

Now, what we are doing here is to try to help *Transport for Edinburgh* test their new app. They are looking for feedback over the next two weeks. Of course, we have all got other things going on as well but **that is their timescale, and it is important to them**.

—

Once you get through that first stage of saying "I'd like to volunteer to be a tester!", you have to register your device[14] with a software distribution tool called *HockeyApp*.

That is fairly easy to do. There are instructions at a given URL[15] which say how to do it.[16] Once that is done, anyone who registers can get a build and download the app onto their phone. But (of course!) you have to do this from your iPhone's Safari browser. If you do it from a laptop, or something like that, you get the `.ipa` file but you can't do anything with it because whichever kind of laptop you have, your laptop will run an operating system, which is different from iOS.

—

The Help page for *HockeyApp* has a question which relates the story of someone who needs to register their phone and does not quite know how to do that. Then, Thomas from *HockeyApp* replies.

> Open `https://install.hockeyapp.net` in Safari on your iPhone, follow the steps, then make sure to sign in so the device is assigned to your account.
>
> Best,
> Thomas

That is the instruction to follow, so you click on the link and you are taken to a web page which talks about registering your device.

When installing software from a web site you should always check to see *where was this web page served from?* Most of them[17] are secure (HTTPS) web pages so you can feel comfortable in doing that[18] but it is always a good idea to look at the URL at the top to make sure that you know where the web page is coming from.

3. Registering your device

You have expressed interest in being a tester. To go further you have to register your device. To register your device tap the button which says "Install". This will create an icon on your home screen allowing quick access to *HockeyApp*[19].

—

[14] At this stage, at the time of writing this note, this is just for the iPhone, but we think that similar apps for other platforms will appear in time.

[15] `http://support.hockeyapp.net/discussions/problems/25566-how-do-i-register-my-iphone`

[16] At the moment, one student in the class has done that.

[17] In the *HockeyApp* workflow.

[18] Or not, as the case may be.

[19] This is a *symbolic link* essentially, linking to a web page.

You are informed that a warning dialogue will appear. You may have to enter your passcode. And that is about that.

—

Then, as you go a little bit further through the workflow, you are invited to do something else, which is to install a *profile*.

(8) The next step is *installing the HockeyApp profile*. You see that HockeyApp is displayed in big, bold letters. Underneath that it says *Bit Stadium*[20]. It is signed by a certifying authority. It is verified. It has got a green tick. You "install this profile to access HockeyApp with your UDID[21]". The Install Profile page contains a *Device Enrolment Challenge* and some more details (where you get the device enrolment challenge and signing certificate with expiry dates[22]).

(9) Then you have to enter your standard passcode which you use to unlock your phone when it has been locked. You do that and you go a bit further on.

(10) Now you get to the stage where you are installing the profile. The screen goes a little bit grey, and some buttons appear at the bottom saying "Install" or "Cancel". If you click "Install" you can install the profile.

(11) Having clicked "Install" you will then receive a success message saying "Profile Installed. Done." This is so that you know what it is that you have done. The message tells you that the profile contains a "Web clip".

Then you go a little further on through the workflow.

You are downloading software onto your device in order to be able to test the app. So, what is the next stage?

(12) The next stage you can probably guess. You always have to make an "account", and always have to sign in so that people will know who it is that is testing the app. It should be the case that different people are testing it, not just one person testing it 100 times for some reason.

So, you make an account (with HockeyApp).

You sign in.

(13) We go to HockeyApp, which allows you to sign in with an existing account on Facebook, Google+, or Microsoft, to save you having to create yet another user ID, and use yet another password. (I signed in with my Google+ account.)

(14) Then I got to the stage where I was presented with a Dashboard and it told me "OK, you seem to have expressed interest in this app called *TFE (Transport for Edinburgh) Talk* from *Lothian Buses PLC*."

There is a big warning sign which says that this is a Beta release: there may be the usual bugs that you find in a beta version of software, there may be more testing to be done ... (but that was why I had signed up!). But, still, this warning about beta-release software is nicely and clearly displayed.

You can look a little bit further there but at that point you are stuck until the developer chooses to add your device.

[20] `https://github.com/bitstadium`

[21] **U**nique **D**device **ID**entifier. See `http://whatsmyudid.com` on how to find yours and read more at `https://theiphonewiki.com/wiki/UDID`

[22] How interesting that it shows you the expiry dates for the signing certificates.

It reminds you again that you cannot install the app until you receive an email letting you know that a build is available.

—

At that stage, what are you left with on your device?

—

You have installed a profile: what else have you got?

You have now got an icon on your home screen which is the HockeyApp icon which allows you to download software from HockeyApp.

—

So, that is the workflow.

Have a think why all of those steps were necessary.

And why were they done in that order?

— ⋄ —

Now, consider this video — `http://youtu.be/bKw2Xtt_QWo` — from *Transport for Edinburgh* showing you what the app will be like once you get it.[23]

The person using the phone in the video is sighted: they are tapping exactly the right place on the screen.

When moving the phone around, it announces nearby stops in a particular direction.

Suppose you are on the bus. The bus has got wi-fi. It will tell you the next stop arrivals. ("Next stop Shrubhill".[24])

And that is the short video clip advertising the app.

—

The availability of this app does not invalidate the creation of other apps for even the same platform because no-one knows how to solve these problems thoroughly. Having more attempts at doing it will be really valuable.

But this is Transport for Edinburgh's current attempt. It's a beta. It's the first time that they have tried something like this.

There can be problems ahead, or there simply may be successes after successes after successes, but in either case I certainly applaud Transport for Edinburgh for trying!

—

Boldly go where none have gone before!

Acknowledgements: Thanks for John McEvoy for providing the link to his app.

[23] Among the many challenges faced by the app include the pronunciation of Scottish place names by a miniaturised American gentleman inside a mobile telephone. The text on the screen reads "25 to Riccarton, 6 mins ⟩" which is served to VoiceOver as "Service 25 to Riccarton. Arrives in six minutes." The miniaturised American gentleman pronounces "Riccarton" as "Rick Art Un" whereas Edinburgh natives pronounce it as "Ricker Ton".

[24] This person is probably on the #25 bus which passes through Elm Row and Shrubhill on its way down Leith Walk. Map: `http://lothianbuses.com/assets/files/r25_130324.pdf`