Mobile Development Primer

Getting from idea to development

Written by Simon Judge, Mobile Developer.
About the Author
Simon Judge is a mobile application developer in the UK with over 20 years commercial IT experience. He has been working in mobile since 1996 when the Windows CE 1.0 beta (Pegasus) appeared. Since then he have worked on Windows Mobile, Symbian OS, S60, UIQ, Java ME, BlackBerry, Android and iPhone. He is an Accredited Symbian Developer.

Simon’s company provides advice, feasibility studies and design/implement applications that add functionality to mobile devices. You can find out more about our consultancy and development services at SimonJudge.com. You can read Simon’s blog at mobilephonedevelopment.com.

History
January 2010 First Edition
March 2010 Second Edition
September 2010 Third Edition

Disclaimer
The accuracy and completeness of information provided and opinions stated herein are not guaranteed or warranted to be correct. Any advice and strategies provided may not be suitable for every circumstance. The author shall not be liable for any loss incurred as a consequence of the use and application, directly or indirectly, of any information presented in this work.

Copyright © 2010 Simon Judge. All rights reserved.
No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, taping, or by any information storage retrieval system, without the written permission of the author.
Introduction
The popularity of mobile, particularly the iPhone, has resulted in many individuals and companies approaching me with ideas for mobile development. Some people aren’t from the mobile ecosystem or even from a software development background. These people ask how they can go about developing a mobile idea. These people often ask the wrong questions, assume the wrong things and generally need a lot of hand-holding to get them to a stage where I can design and develop their mobile application. Often, these people have previously contacted other mobile developers but have come away disillusioned and disappointed.

These tips will hopefully help you see mobile development services from a developer’s point of view.

This primer is suitable for a range of people…

- Entrepreneurs and business owners considering mobile development
- Students learning about mobile
- Analysts wanting to understand mobile development
- Investors wishing to learn about the mobile development process

This primer is an amalgamation of information I have written on my blog, general information I have provided within feasibility studies and advice I have given to companies and individuals. It provides a single reference to which I point people and hopefully provides help to a wider range and number of people.

This isn’t an introduction to programming. There are sufficient books and web sites that cover programming. Instead it takes you from the point you have an idea, through choosing one or more platforms, through choosing a mobile developer to agreeing how to work with a developer.
Know what you want doing
A developer will ideally need a specification for what you want doing. Without a specification you are opening yourself (and the developer) to problems due to misunderstandings. If a specification isn't available then you will probably have to ask the developer to create one as an initial task. Even if a specification isn't initially available, before you approach a developer you should have a bullet point list of things that you want the application to do.

There’s a saying that ‘the devil’s in the detail’. In mobile, this is often the case. All too often companies make their applications too complex while also forgetting to include vital features.

From experience, here are some things that are often forgotten:

- The user changing their SIM and hence their data connection or telephone number
- What happens when the user is roaming (e.g. high data costs)
- Diagnosing end-user problems via error logs
- Maintaining or deleting data when a new version is installed
- Defining what happens when the phone runs out of memory or storage
- Defining what happens if the application gets interrupted (e.g. incoming SMS or phone call)
- Defining what happens when only part of a multiple-step operation succeeds (e.g. data atomicity)
- Supporting foreign language variants of the phone (particularly different directory names)
- The user reinstalling from memory to memory card
- The user changing their registration details (email, name, address etc)
- The user opting out of a service (e.g. deleting data)

Some projects are simple at the start and end up more complex due to ‘real world’ issues. Many of these issues are the same as those experienced by people creating applications for the PC or server. The ‘real world’ sometimes needs to cater for some or all of:

- Multiple time zones
- Multiple languages
- Multiple locales (e.g. date/time formats)
- Multiple encoding schemes for data
- Multiple preferred ways of paying across countries
- Multiple currencies
- Multiple country codes
- Multiple network operators
- Multiple phone number formats
- Multiple input types

Finally, if your project is client-server, also think about what needs to be done on the server side (and by who). Some mobile developers can also take on this kind of work.
Know what you want to do is possible

Mobile is technically complex. Almost all non-trivial projects need some kind of research to work out what's possible on what platform and assess the areas of most risk. When a developer is asked to quote on a complex project, they can do one of several things:

- Fully research the project. Only large software houses with large overheads and large fees can afford to do this.
- Assume it's all possible, leave the risk with you without telling you and only let you know when they eventually reach something that's not possible.
- Go quiet and lose interest in quoting for the project because too many things are uncertain without significant time spent on research.
- Offer to do a feasibility study for a fee.

I usually recommend and take the fourth route. Feasibility studies are a great way for the developer and client to get to understand one another, evolve the project based on past experience, and build trust.

Know what platforms you want developing

Too many people approach me asking for an application to work on 'All mobile phones'. This isn't possible. Work out the exact platforms and/or specific phones. The choice of platform usually depends on:

- Geography - Some platforms are more popular in some countries than others
- Capability - Some platforms can't do some things
- Demographics – e.g. Some platforms are used more by enterprise than consumers
- Capability - What can you program and have a passion for?
- Market - Some platforms have better routes to market
- Timing - Some platforms are more for the future than for now, some are for now, others are re-inventing themselves

You might like to look at my list of free Market Research to help with some of these issues.

If you are a company with existing customers and are extending your services to mobile then also consider how supporting just one or two mobile platforms might alienate customers who don't have those platforms.

Within platforms, you need to know what versions you need to support. Supporting older ones probably won't be economic. Here's a list of what I would ask for if I were commissioning an application to run on recent (last few years’) phones:

- Symbian - S60 3rd Edition and 5th Edition
- iPhone - All
- Android - All
- Windows Mobile - 5.0 and later. Windows Mobile Standard (non-touch) and Professional (touch)
- Java ME - MIDP 2
- BlackBerry - 4.2 and later
Choose a developer

The remainder of this document is applicable if you are outsourcing development. Outsourcing provides a flexible, scalable workforce while providing expertise and experience that can affect the outcome of your project. You can resource projects quickly and scale up and down as requirements change. Most outsourcing provides people who are results driven and need less motivating than permanent employees. Good mobile developers will be skilled in asking comprehensive, in-depth questions resulting in a higher chance of a better solution.

Contrary to expectation, outsourcing often leads to greater continuity than with permanent staff whose attachment ends when they move on. Business minded mobile developers remain loyal to the interests of the client, even after the project has been completed.

There are five main types of mobile developer:

**Large generic software consultancy**

Large software houses and consultancies will rarely be interested in mobile software projects unless they are part of a much larger project. Mobile, by its very nature, requires a relatively small amount of software to be developed which is usually of too low a financial value to make it worth their while.

Incidentally, I started my career at Logica, which was one of the largest independent UK software development houses.

**Small mobile software house**

There is a growing number of mobile software companies that tend to have between 5 and 50 employees. An advantage of these companies is that they usually specialize in mobile. They also have a flexible workforce in that if you need something done quickly or a developer becomes ill then more people can be put on the project.

A disadvantage is that you tend to speak with someone, usually a salesman or manager, who doesn’t end up doing the work. In some circumstances this can lead to variations in expectation. Consequently, always ask to speak to or meet the person who will actually be doing the work so that you can gauge their experience, understanding, communication skills, attitude and motivation.

Small mobile software companies tend to be more expensive (generally about 1.5x the cost of freelancers) due to the overheads of running a larger company. Costs are considered in more detail in the next section.

**Freelance**

The advantage of freelancers is that the person you initially speak with is the one who will do the work. You can directly assess their skills, experience, attitude and communication skills.

The disadvantage of freelancers is that they can be of variable reliability. The problem is that some don’t have the self-discipline not to take on too much work and consequently end up being vague with regard to timescales. When approaching freelancers, ask how long the project will take and when they can start. Ask how much other work they plan to do concurrently.
**Offshore**

Both small mobile software houses and freelancers can be offshore, usually in Russia, India or increasingly in China. The advantage is obviously cost that can be up to 1/5 that of developers in Europe or N America.

Offshore is a great solution if you:

a) Have time to manage the relationship. Having asked past clients and people at mobile conferences, I have found that it’s generally acknowledged that offshore projects take more management.

b) Know exactly what you want and that your project is 100% possible. The reason I say this is because I see projects where offshore developers approach me when they have got stuck. They sometimes take on impossible projects without fully assessing whether they are technically feasible. UK companies have also approached me to take over their project when their offshore project has gone wrong.

Obviously, there are many offshore projects that go well. It’s all about risk vs cost. It’s knowing you are willing to put in more time managing the project and ensuring, at inception, that your project is feasible.

**Student**

Some projects just don’t have the budget for professional development. One solution is to employ a student who may even be able to incorporate part or all of your project into their coursework as a vocational or holiday activity.

Beware that a student may be very unpredictable and might drop the project at any time. Also, the project probably won’t be implemented in an optimal or professional way. Nevertheless, it can be a great way to get a prototype working prior to gaining venture capital and re-developing.

For all Types of developer

With *any* type of developer, always make sure you communicate with the person who will be actually doing the work. Good communication is very important. Their skill, previous experience, attitude and passion for mobile will greatly influence the success of your project.

Also look for case studies on developers’ web sites. Developers should be able to put you in touch with past clients who should be more than happy to have an informal telephone chat.
Work out roughly how much it will cost

Mobile application development costs depend on the size of the project and the cost of the developer. An expensive developer doesn't necessarily cost more because they should be more experienced and will be able to do things quicker and more optimally. Optimal solutions include not trying to implement things that can't be done, writing for re-use, intelligent design so that things don't have to be re-factored later and forward planning so things don't get missed.

Most projects take between a few weeks and a few months per platform. Looking at my past projects, the average time to create a non-trivial mobile application is about 6 to 8 weeks. In fact, if your project is larger than that then you are probably trying to do too much for your first iteration. You might be better doing less and improving the application later based on user feedback and ideas – that will nearly always be different and better than if you had try to create the ideas yourself.

Professional European and N American developers charge between £300 (about $450, €330) and £650 (about $980, €720) per day. The variation in price depends on experience, whether freelance or small mobile company and how much work a developer has on (the developer's popularity).

Hence, the cost of an average project will be of the order of tens of thousands of pounds, dollars or euros.

Prove you can pay

Some mobile developers will have had problems with bad payers in the past. Explain how the project will be funded. Offer to provide a purchase order that commits you to the work. Some developers may do credit checks. If you are a new or unknown company then you may be asked for an initial deposit.

Agree payment terms

There are generally two ways to pay for mobile development. Fixed price and Time and Materials (T&M). Fixed price may seem advantageous but the problems start to happen when the application veers away from the original specification (as it always does!). Conversely, Time and Materials on a daily or hourly rate might be seen as a way for the developer to charge as much as they like.

I prefer a mix between fixed price and T&M. I quote how long it will take to develop and multiply that by my daily rate. If the project takes more time then that is my risk. Incidentally, I also usually work outside normal hours (free of charge) to meet the agreed schedule. If the client changes the specification, which demands additional work, then they know the daily rate that will be applied.

Know what will be delivered when

Beware of developers that take on a project and say they will deliver in several months' time. How will you know what is being developed is actually what you wanted? A better approach is to get the developer to create a plan with weekly or fortnightly deliveries so that you can see what's being done as work progresses. This way you can intervene before it's too late. Also, it allows payment to the developer be phased.

When planning development, I split the tasks into small related groups of functionality that take about a week. I then release what I have done every week. This allows my customer to see what I am doing that, in turn, allows them to…
• Share in decisions, problems and achievements
• Ensure that risky items are done upfront so that any impact is known earlier
• Change the way something has been implemented
• Assess if the project is going to plan
• Provide measurable activity against which payment can be made
• Help informally test what’s been done
• Provide early alpha and beta versions to their potential customers or partners

Releasing often focuses the mind and encourages the creation of code that’s release quality first time rather than relying on things to be deferred until later that sometimes never get done.

The alternative, releasing everything at the end is usually a recipe for disaster. It, more often that not, results in a discontinuity between customer and developer expectations.

**Agree a contract**

Have yourself or the developer draw up a contract. Ensure it includes:

• Payment terms
• What is to be delivered when
• Who owns the source code
• Whether any open source code will be used and the implications on licensing the end product
• Confidentiality terms (if not already in a non disclosure agreement)